

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»

**ПРАКТИКУМ ПО ДИАГНОСТИКЕ КОМПЬЮТЕРА  
С МОДУЛЕМ ТЕСТИРОВАНИЯ ПЕРИФЕРИЙНОГО  
ОБОРУДОВАНИЯ**

Выпускная квалификационная работа  
по направлению подготовки 44.03.04 Профессиональное обучение  
(по отраслям)  
профилю подготовки «Информатика и вычислительная техника»  
специализации «Компьютерные технологии»

Идентификационный номер ВКР: 507

Екатеринбург 2017

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующая кафедрой ИС

\_\_\_\_\_ Н. С. Толстова

« \_\_\_\_ » \_\_\_\_\_ 2017 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**ПРАКТИКУМ ПО ДИАГНОСТИКЕ КОМПЬЮТЕРА**  
**С МОДУЛЕМ ПЕРИФЕРИЙНОГО ОБОРУДОВАНИЯ**

Исполнитель:

обучающийся группы № ЗКТ-412с

Р. Ф. Миннигалиев

Руководитель:

ст. преподаватель

С. В. Чёнушкина

Нормоконтролер:

Б. А. Редькина

Екатеринбург 2017

## РЕФЕРАТ

Пояснительная записка к выпускной квалификационной работе выполнена на 80 страницах, содержит 58 рисунков, 3 таблицы, 29 источников информации, 2 приложения.

Ключевые слова: C#, Microsoft visual studio 2010 C#, практикум, диагностика, тестирование, периферийное оборудование.

**Объектом исследования** является процесс обучения основам диагностики и выявления неисправностей устройств компьютера и периферийного оборудования.

**Предметом исследования** являются лабораторные работы по диагностике компьютера с возможностью тестирования внутренних устройств и периферийного оборудования.

**Целью работы** является разработка практикума по диагностике компьютера с модулем тестирования периферийного оборудования.

В соответствии с поставленной целью в работе определены следующие задачи:

1. Проанализировать данные в области неисправностей и диагностики компьютерного оборудования.
2. Подготовить лабораторные работы на основе использования программных средств компьютерной диагностики.
3. Разработать модуль диагностики периферийного оборудования.
4. Реализовать интерфейс и наполнение практикума с возможностью запуска средств диагностики.
5. Протестировать продукт на различных компьютерных архитектурах.

# СОДЕРЖАНИЕ

Введение.....	4
1 Теоретические аспекты в области диагностики компьютера.....	6
1.1 Краткий обзор неисправностей .....	6
1.1.1 Основные категории неисправностей компьютера .....	6
1.1.2 Основные методы диагностики внутренних устройств.....	8
1.1.3 Обзор программ по диагностике .....	14
1.2 Обзор документации по диагностике и технологиям разработки .....	19
1.2.1 Обзор литературных источников .....	19
1.2.2 Обзор интернет-источников.....	21
1.2.3 Вывод по первой главе .....	24
2 Описание практикума и модуля диагностики.....	25
2.1 Педагогический адрес.....	25
2.2 Общие требования по созданию лабораторных практикумов .....	26
2.3 Выбор средств реализации практикума и модуля .....	30
2.4 Интерфейс, структура и навигация практикума .....	30
2.5 Описание собственного модуля диагностики оборудования.....	33
2.6 Описание модулей работы с программами диагностики.....	43
2.7 Описание лабораторных работ .....	49
2.8 Рекомендации для обучаемых и преподавателя .....	60
2.9 Апробация практикума и модуля .....	61
Заключение .....	62
Список использованных источников .....	64
Приложение 1 .....	67
Приложение 2 .....	69

## ВВЕДЕНИЕ

Современный мир – мир компьютеров и компьютерной техники. Компьютера уже не являются предметом роскоши, а стали обыденным инструментом для выхода в сеть интернет, для работы, учёбы, просмотра фильмов, прослушивание музыки и самое популярно применение – играть в видео игры. А что происходит с компьютером во время игр?

Системные блоки компьютеров имеют разные вычислительные мощности, и чем выше цена сборки системного блока, тем и выше его мощность. Во время игр, на компоненты компьютера ложится большая нагрузка, что со временем приводит к ошибкам и сбоям системы.

Сбой может произойти в любой момент и в момент работы системы и при его включении. Некоторые проблемы имеют быстрое решение, и возить их в центры по диагностике и ремонту компьютерной техники будет накладно для домашнего кошелька. Каждый пользователь персонального компьютера может сделать диагностику своего инструмента в домашних условиях, используя специально разработанное программное обеспечение.

Диагностическое программное обеспечение чрезвычайно необходимо в том случае, если система начинает сбоить или если вы модернизируете ее, добавляя новые устройства. Даже когда вы пытаетесь выполнить простую операцию (например, установить новую плату) или ищите неисправность в аппаратуре, приведшую к сбою или «зависанию» системы, вам необходимо знать о компьютере больше, чем написано в прилагаемой к нему инструкции. Диагностические программы позволяют проверить работу, как всей системы, так и отдельных ее узлов. Естественно, при эксплуатации системы меню необходимо регулярное техническое обслуживание. Именно это и служит залогом нормальной работы компьютера.

Диагностика поможет существенно сохранить сбережения в доме. Наровне с диагностикой идёт такое понятие как техническое обслуживание.

Техническое обслуживание – мероприятия профилактического характера, проводимые систематически, принудительно через установленные периоды, включающие определённый комплекс работ. Несмотря на великое многообразие периферийных устройств ПК, все они взаимодействуют с процессором и оперативной памятью примерно одинаковым образом.

Таким образом, специалисты в области компьютерных технологий должны обладать умениями в области диагностики неисправностей ПК и периферийного оборудования. Это могут быть профессиональные специалисты по ремонту компьютеров и студенты, обучающиеся по профилизации «Информационные технологии».

**Объектом** исследования является процесс обучения основам диагностики и выявления неисправностей устройств компьютера и периферийного оборудования.

**Предметом** исследования являются лабораторные работы по диагностике компьютера с возможностью тестирования внутренних устройств и периферийного оборудования.

**Целью** работы является разработка практикума по диагностике компьютера с модулем тестирования периферийного оборудования.

В соответствии с поставленной целью в работе определены следующие **задачи**:

1. Проанализировать данные в области неисправностей и диагностики компьютерного оборудования.
2. Подготовить лабораторные работы на основе использования программных средств компьютерной диагностики.
3. Разработать модуль диагностики периферийного оборудования.
4. Реализовать интерфейс и наполнение практикума с возможностью запуска средств диагностики.
5. Протестировать продукт на различных компьютерных архитектурах.

# **1 ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В ОБЛАСТИ ДИАГНОСТИ КОМПЬЮТЕРА**

## **1.1 Краткий обзор неисправностей**

### **1.1.1 Основные категории неисправностей компьютера**

Многих неисправностей можно избежать, используя постоянную профилактику компьютера.

Профилактика компьютера заключается в следующем:

1. Очистка и продувка от пыли (материнская плата, блок питания, кулера на процессоре, видеокарте и в блоке питания и т.д.);
2. Прочистка контактов на видеокарте, оперативной памяти;
3. Удаление старой и нанесение новой термопасты на процессоре, видеокарте, чипсет материнской платы.

Так же стоит помнить о том, что пыль является главной виновницей большинства поломок компьютера.

Неисправности компьютеров можно разделить на две категории:

1. Неисправности, связанные с выходом из строя программного обеспечения.
2. Неисправности, связанные с выходом из строя комплектующих компьютера.

В первом случае происходит неправильная работа программы, недоступность или отсутствие некоторых функций и т.д. Устранение неисправностей связанных с программным обеспечением решается простой переустановкой неработающей программы.

Во втором случае всё сложнее. Нужно прибегать к разным тестовым программам, позволяющим выявить неисправный элемент. Но тут тоже есть

свои подводные камни. Компьютер может вообще не запускаться, тогда и тестирования никакого не будет.

Неисправности можно разделить на несколько пунктов:

Компьютер не включается. Когда говорится «не включается», то это означает, что при нажатии на кнопку включения не происходит замыкания контактов. Не загораются лампочки на передней панели системного блока, кулера не крутятся и т.д.

Здесь может быть три неисправности:

- блок питания;
- материнская плата;
- кнопка включения на передней панели корпуса.

Компьютер включается, но изображение не выводится на монитор. Это означает, что лампочки загораются, кулера крутятся, но изображения на мониторе нет. Это в том случае, если знаем точно, что монитор исправен.

Здесь может быть три неисправности:

- блок питания;
- материнская плата;
- видеокарта;
- оперативная память;
- центральный процессор.

Компьютер включается, изображение появляется, система загружается, но работает нестабильно. Система постоянно виснет, либо вдруг ни с того ни с сего начинает перезагружаться или при работе вдруг пропадает видеосигнал и монитор показывает чёрный экран, либо появляются артефакты на мониторе.

Здесь может быть три неисправности:

- блок питания;
- материнская плата;
- оперативная память;
- видеокарта;



- центральный процессор.

Система загружается, работает вроде нормально, но во время работы проявляется какой-либо дефект. Это означает, что дефект плавающий и явно себя не показывает. Система вполне стабильно может работать довольно долгое время и проявиться может непредсказуемо.

Здесь может быть три неисправности:

- блок питания;
- материнская плата;
- оперативная память;
- видеокарта;
- центральный процессор.

В этом случае используются все вышеуказанные способы:

- прочистить от пыли;
- визуальные дефекты, температурные режимы;
- тестирование программными средствами (оперативную память, видеокарту, центральный процессор).

Правда, чтобы выявить плавающий дефект, тестировать приходится очень и очень долго. Иногда это занимает до нескольких дней.

### **1.1.2 Основные методы диагностики внутренних устройств**

#### **Диагностика состояния блока питания (БП)**

Множество отказов операционной системы, сбоев и различных проблем связано с тем, что в ПК работает некачественный, устаревший или перегруженный БП. Плохой БП способен ухудшить работу даже самых дорогих комплектующих

Если светодиод «Power» на лицевой панели корпуса не светится и при этом вентилятор в блоке питания не работает, то отключите от сети компьютер, откройте корпус системного блока, отсоедините разъемы питания от

системной платы, флоппи-дисковод, CD-ROM, винчестера. Подсоедините БП к любому дисководу, так как совсем без нагрузки его включать нельзя.

Можно также проверить тестером соответствие напряжений и уровень сигнала Power Good.

Ниже в таблице 1 показан разброс выходных напряжений БП.

Таблица 1 - Выходные напряжения блока питания

Номинальное напряжение, В	Допуск отклонений, %	Минимальное напряжение, В	Максимальное напряжение, В
+12	5	+11.4	+12.6
+5	5	+4.75	+5.25
+3.3	5	+3.14	+3.47
-12	10	-10.8	-13,2
+5(+5VSB)	5	+4.75	+5.25

Чтобы ПК работал стабильно и надежно, в нем должен быть установлен качественный, достаточно мощный БП с хорошей защитой.

Чтобы БП прослужил дольше, необходима профилактика. Главный враг БП – это пыль. С потоком воздуха она попадает внутрь и, оседая на деталях БП, ухудшает теплоотдачу. Рекомендуются периодическая чистка от пыли как снаружи, так и изнутри.

Еще одна серьезная опасность – насекомые. Им нравятся «теплые» места, они часто устраиваются внутри БП. Попадая между проводниками, насекомое вызывает КЗ.

### **Диагностика состояния материнской платы**

Внимательно осмотрите материнскую плату. Если на ней есть механические повреждения, посмотрите, не нарушены ли токопроводящие соединения. Возможен скол на краях. Если повреждения коснулись только текстолита – изоляционной основы, на чем смонтирована плата, то, скорее всего, ничего страшного не произошло. Если повреждены контакты, токопроводящие дорожки, то материнская плата, вероятнее всего, перестанет работать.

Если в случае перегрева вышел из строя какой-либо элемент, обратитесь в сервисный центр.

Плату, на которой есть глубокие трещины, царапины, скорее всего, придется заменить, так как могут быть задеты несколько слоев платы. Надежда исправить дефект существует лишь при повреждении верхнего слоя.

Изучение звуковых и текстовых сигналов BIOS также поможет определить некоторые неисправности МП. Возникшая неисправность определяется на основании полученного текстового или звукового сообщения. Эти сообщения генерируются процедурой POST BIOS во время начальной загрузки компьютера, сразу после его включения. Способ диагностики МП с помощью сигналов BIOS позволяет более точно определить неисправность.

Ниже в таблице 2 и в таблице 3 приведены сигналы BIOS с их расшифровкой.

Таблица 2 – Сигналы AMI BIOS

Сигналы	Проблемы
нет сигнала	Проблемы питания
2 коротких	Ошибка тактового генератора памяти
3 коротких	Ошибка в первых 64kb памяти
4 коротких	Ошибка системного таймера
5 коротких	Неисправность процессора
6 коротких	Неисправен контроллер клавиатуры
7 коротких	Неисправность материнской платы
8 коротких	Неисправность VGA карты
9 коротких	Ошибка теста BIOS
10 коротких	Ошибка записи в CMOS
11 коротких	Неисправность КЭШ памяти материнской платы
1 длинный + 2 коротких	Неисправность VGA карты
1 длинный + 3 коротких	Неисправность VGA карты
1 длинный + 8 коротких	Монитор не подсоединен

Таблица 3 – Сигналы AWARD BIOS

Сигналы	Проблемы
2 коротких	Ошибки в CMOS Setup or motherboard
3 коротких	Неисправен контроллер клавиатуры
1 длинный+1 короткий	Ошибка памяти
1 длинный+2 коротких	Неисправность VGA карты
1 длинный+3 коротких	Ошибка инициализации клавиатуры
1 длинный+9 коротких	Ошибка чтения ROM
короткий, recurrent	Неисправность источника питания
длинный, recurrent	Проблема памяти
непрерывный	Неисправность источника питания

Еще возможно изучение сигналов BIOS с помощью специальной диагностической платы. Этот способ также основан на сообщениях, генерируемых процедурой POST BIOS. проведения диагностики вам понадобится только диагностическая плата.

### **Диагностика состояния центрального процессора (ЦП)**

Маловероятно, что выйдет из строя ЦП, намного чаще «вылетает» МП. Но все же, если это произошло, например, в результате перегрева или разгона, – остается лишь установить новый и постараться исключить подобную ситуацию в дальнейшем.

Основные причины, которые могут привести к выходу из строя ЦП, это:

- перегрев;
- серьезная поломка МП;
- замыкание контактов (например, из-за попадания влаги);
- разгон, запуск процессора на более высокой частоте, на которую он не был рассчитан;
- сильный скачок напряжения.

Самое главное – контролировать температуру процессора, чтобы не допустить его перегрев, который сказывается на общей продолжительности работы ЦП. Проследите за показаниями датчика температуры, когда система

простаивает и когда она сильно загружена. Отклонение от этих величин может предупредить о возникновении неполадки.

Если температура процессора меньше 35 °С, то беспокоиться не о чем. 35–45 °С также в пределах нормы. До 60 °С допускается нагрев процессоров, особенно мощных или же при выполнении большого объема вычислений. Нагрев до 60–65 °С говорит о необходимости проверить систему и выяснить причину этого. Если датчик сигнализирует о превышении 70 °С, то следует сразу выключить компьютер и разобраться в причинах появления перегрева.

Старайтесь периодически чистить ПК от пыли и грязи, которая скапливается внутри, и особенно на вентиляторе, радиаторе, вокруг сокета. Следите за вентиляцией корпуса.

### **Диагностика состояния оперативной памяти**

Если при установке операционной системы (лучше с лицензионного дистрибутива) возникают непредвиденные проблемы (зависания, сообщения об ошибках программы установки или памяти, не устанавливаются «родные» драйверы видеокарт и других адаптеров и т.д.), то, скорее всего, они связаны именно с проблемами ОЗУ. Обычно замена модуля DIMM решает возникшие проблемы.

О неисправности оперативной памяти сообщит BIOS подачей соответствующих сигналов.

Извлеките модуль памяти, промойте спиртом контакты модуля и верните его обратно. Иногда чистят контакты ластиком, но такой подход не рекомендуется. Небольшой кусочек ластика может случайно попасть в разъем и заизолировать контакт.

Случается, что модуль памяти не до конца вставлен в слот. При этом ПК может даже заработать, но в дальнейшем с ним возникнут проблемы. Всегда проверяйте правильность установки модулей памяти.

Если у вас возникли неполадки ПК и есть подозрение на модуль памяти, проверьте его на другом – исправном ПК. Также неплохо проверить свой ПК, если есть исправная память.

### **Диагностика состояния накопителя на жёстком магнитном диске**

Неполадки с накопителями на жестком магнитном диске происходят, как правило, чаще, чем с остальными устройствами. Повреждается диск при встряске, причем во время его работы. Поэтому старайтесь не двигать диск, системный блок во время работы. Не всегда проблемы заключаются в его физической поломке.

Если слышны громкие щелчки, стук внутри HDD, скорее всего, звук вызван ударом головки о пластину.

В домашних условиях разбирать HDD, менять магнитные головки явно не стоит. Так как необходимо специальное оборудование (хотя при большом желании это оборудование можно сделать и самому). Разбирать HDD нужно в «стерильном» помещении, то есть с минимальным содержанием пыли в воздухе. Лучше обратиться в сервис-центр.

Причиной подобного сбоя накопителя служит и «загибающийся» блок питания.

При этом есть большая вероятность того, что он выйдет из строя раньше, чем вы полагаете.

Различные необычные звуки, издаваемые накопителем, предупреждения на экране системы о том, что не удастся прочитать или записать данные, а также предупреждения во время загрузки системы о том, что есть неисправность HDD. Если списать данные не удалось, то причина неисправности может заключаться в том, что вышла из строя плата контроллера или неисправен блок головок – head-disk assembly (HAD).

### **Диагностика состояния видеокарты**

Неисправности видеокарты – достаточно распространенное явление.

Если на мониторе нет изображения, необязательно неисправным окажется именно он. Вначале следует проверить исправность монитора, например, подключив его к другому системному блоку или временно заменив видеокарту на заведомо исправную.

При отсоединении сигнального кабеля монитора на его экране обычно высвечивается сообщение об отсутствии видеосигнала, что даст основание сделать предварительный вывод о его работоспособности.

Если вы убедились, что МП, процессор и ОЗУ работают (динамик системного блока подает соответствующий сигнал), то не исключено, что неисправна видеокарта. Ее лучше всего проверить на другом ПК.

Если на видеокарте установлен вентилятор и он со временем «забился» пылью, перестал вращаться, то его лучше заменить новым, впрочем, он разбирается, необходимо прочистить от пыли и смазать.

С видеокартой следует обращаться предельно аккуратно.

Если контакты на видеокарте запыленные, грязные, протрите их спиртом, и затем вставьте карту и проверьте.

Если визуальный осмотр показал, что есть механическое повреждение – отсутствует элемент платы, например, случайно выдернут конденсатор или транзистор либо отверткой процарапаны проводники, то нужно восстанавливать нарушения.

### **1.1.3 Обзор программ по диагностике**

Для диагностики оборудования компьютера используются различные количество программ, написание как для одного оборудования, так и для всех компонентов в одной программной оболочке.

Рассмотрим некоторые программы и их предназначения:

1. AIDA64 – условно-бесплатная программа, с 30 пробными днями, мощный инструмент для диагностики и тестирования персонального компьютера. Программа предлагает пользователю широкие возможности для разгона и диагностики аппаратной составляющей ПК, стресс-тестирования, а также контроля температуры датчиков. Программа обладает набором уникальных тестов процессора, оперативной памяти и жестких дисков (в том числе твердотельных).

Демонстрацию вывода информации о системе компьютера показано на рисунке 1.

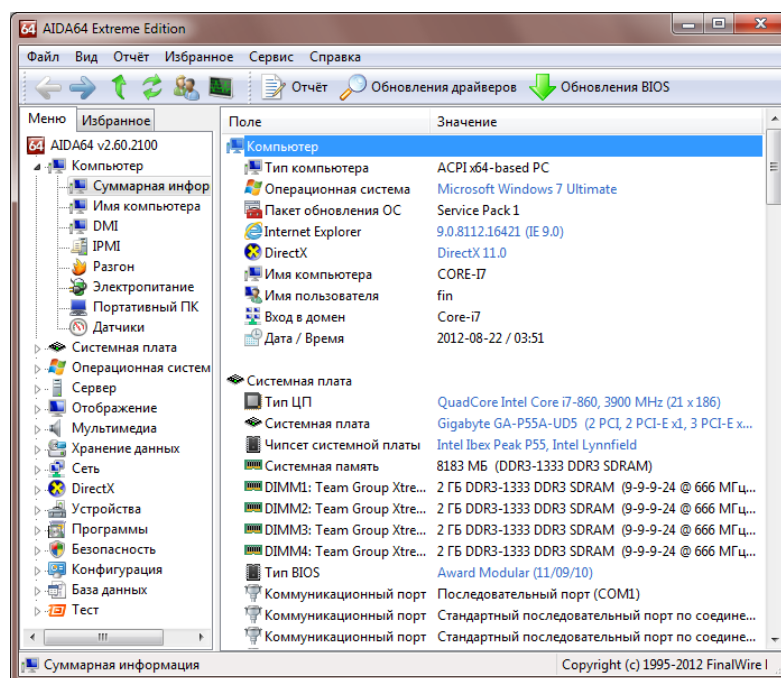


Рисунок 1 – AIDA64 информация о компьютере

2. GPU-Z – бесплатно распространяемая программа, разработанная, чтобы предоставить жизненно важную информацию о видеокарте и графическом процессоре.

Образцы работы продемонстрированы на рисунке 2 и на рисунке 3.

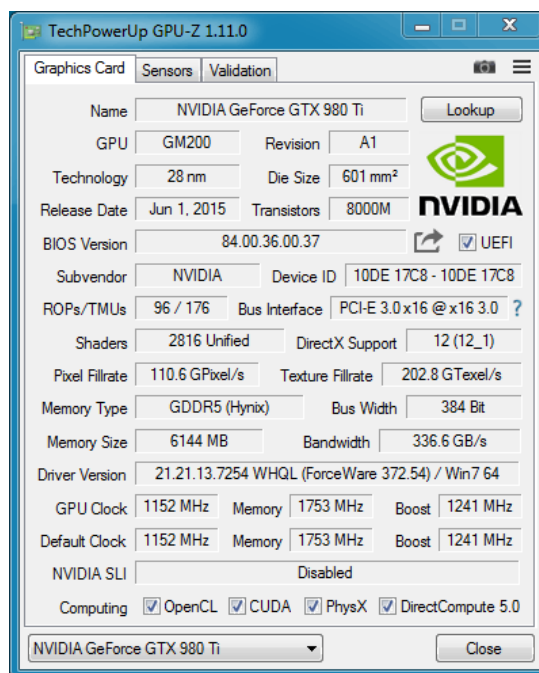


Рисунок 2 – GPU-Z главное меню



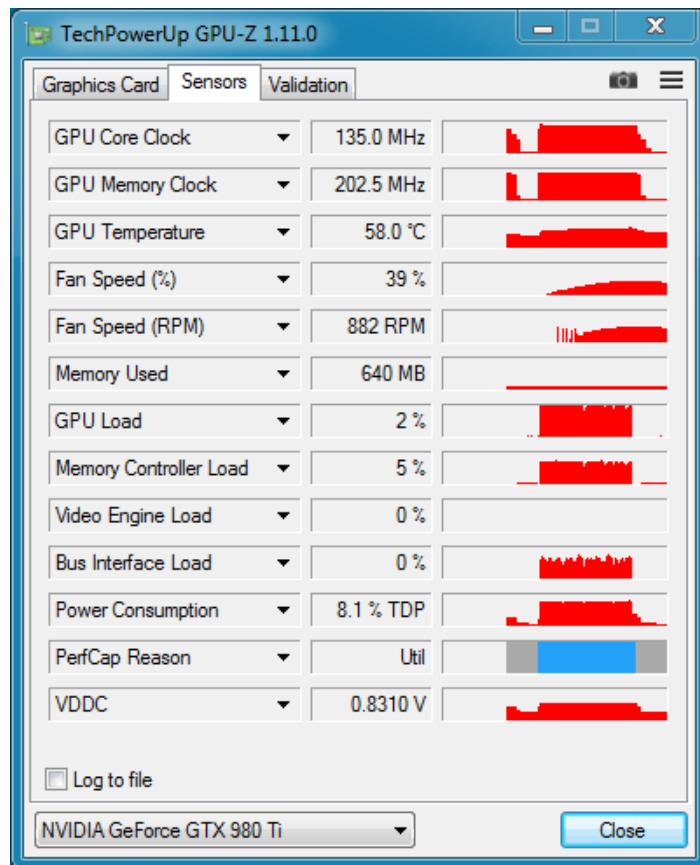


Рисунок 3 – GPU-Z сенсоры

3. CPU-Z - это бесплатная программа, которая собирает информацию о некоторых основных устройствах вашей системы:

- название процессора и количество, название, процесс, пакет, кэш уровня.
- системная плата и набор микросхем.
- тип памяти, размер, тайминги и спецификации модуля (СПД).
- в реальном масштабе времени измерение внутренней частоты каждого ядра, Частота памяти.

Пример работы приложения на базе Intel Core I7 4790K показан на рисунке 4.

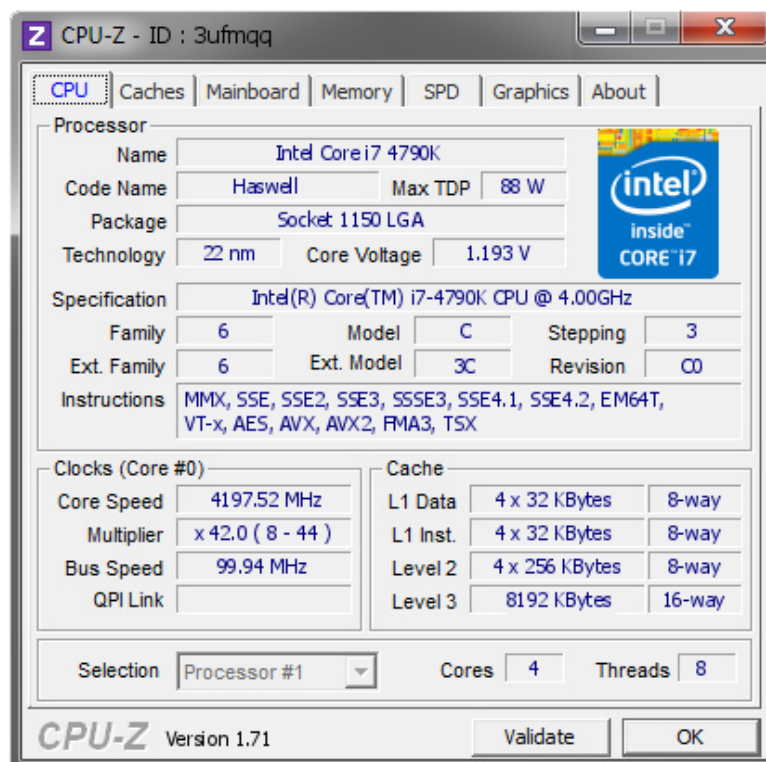


Рисунок 4 – CPU-Z главное меню

4. MHDD – бесплатная программа для диагностики жёсткого диска, проверка SMART и на «битые» сектора. Программу необходимо запускать со съёмного носителя до запуска операционной системы.

Пример работы MHDD продемонстрирован на рисунке 5.

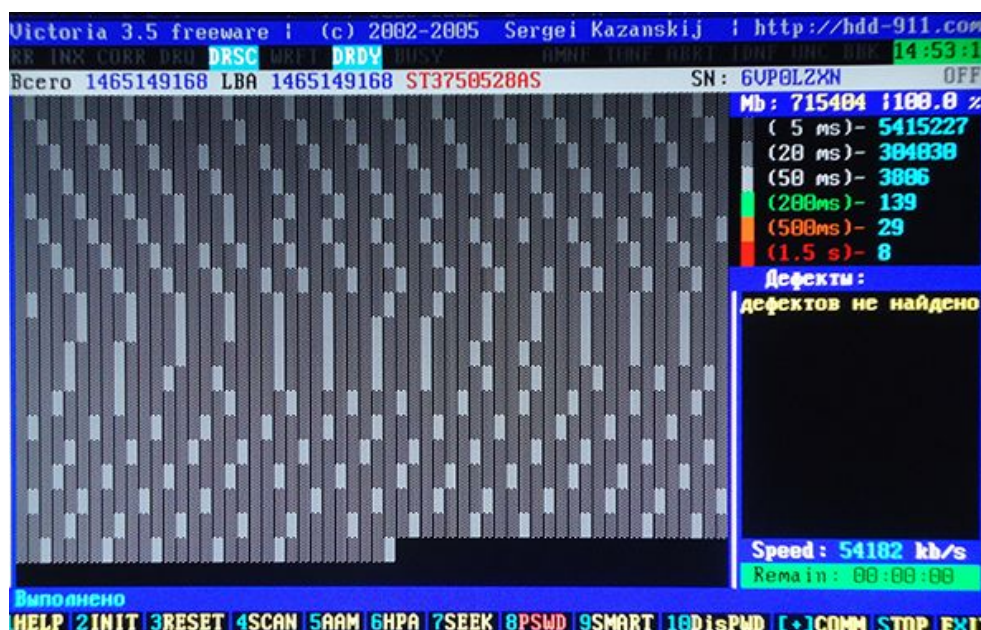


Рисунок 5 – Пример работы MHDD

5. Memtest86 – бесплатная программа для диагностики модулей оперативной памяти на ошибки. Программу необходимо запускать со съёмного носителя до запуска операционной системы.

Пример работы memtest86 с выявленными ошибками памяти (см. рисунке 6).

```

Memtest86+ v2.01 | Pass 54% | *****
Athlon 64 X2 2813 MHz | Test 48% | *****
L1 Cache: 128K 23056 MB/s | Test #6 [Moving inversions, 32 bit pattern]
L2 Cache: 1024K 5728 MB/s | Testing: 2048M - 3583M 4094M
Memory : 4094M 2970 MB/s | Pattern: ffff7fff
Chipset : AMD K8 (ECC : Disabled)
Settings: RAM : 401 MHz (DDR803) / CAS : 5-5-5-15 / DDR-2 (128 bits)

WallTime  Cached  RsvdMem  MemMap  Cache  ECC  Test  Pass  Errors  ECC Errs
-----
3:56:37  4094M      84M  e820-Std  on  off  Std    3      8      0

Tst  Pass  Failing Address          Good      Bad      Err-Bits  Count Chan
-----
6    3    00011d22f60 - 285.1MB  efffffff  efefefef  00100000    1
6    3    00011d22f50 - 285.1MB  feffffff  fefefeff  00010000    2
6    3    00011d22f40 - 285.1MB  ffffffff  ffffffff  00100000    3
6    3    0007cb66f40 - 1995.3MB  efffffff  efefefef  00100000    4
6    3    0007cb66eb0 - 1995.3MB  feffffff  fefefeff  00010000    5
6    3    0007cb66ea0 - 1995.3MB  ffffffff  ffffffff  00100000    6
6    3    0007cb66e90 - 1995.3MB  ffffffff  ffffffff  00010000    7
6    3    0007cb66e80 - 1995.3MB  ffffffff  fefefeff  00100000    8

(ESC)Reboot (c)configuration (SP)scroll_lock (CR)scroll_unlock

```

Рисунок 6 – Пример работы memtest86

Данные программы работают с отдельными компонентами компьютера, и интерфейс бывает, не понятен обычному пользователю, потому что с программами нет описания работы с ними. Лабораторный практикум, в лабораторных работах, напротив помогает пользователю понять интерфейс программ, размещённых в «модуле диагностики», и методы работы с ними. Модуль диагностика содержит комплекс программ для диагностики всего оборудования компьютера, что даёт ему преимущество перед остальными программами по диагностике.

## **1.2 Обзор документации по диагностики и технологиям разработки**

### **1.2.1 Обзор литературных источников**

Авдеев В.А. в своей книге «Организация ЭВМ и периферия с демонстрацией имитационных моделей» [1] описывает проблемы развития инновационных образовательных технологии, включает детальное описание и компьютерное моделирование работы компонентов ЭВМ и периферии: устройств управления и прерывания, памяти, жёстких дисков, принтеров, сканеров, видеосистем и т.д.

Наглядное обучение, предлагаемое в книге, может быть использовано преподавателями для проведения лекционных занятий в увлекательной форме.

В книге В.В. Бахтизина, Л.А. Глухой «Технология разработки программного обеспечения» [3] рассматриваются основные понятия и определения в области технологий разработки Программных средств (ПС) и систем. Кратко описываются процессы жизненные циклы (ЖЦ) ПС и систем, регламентированные стандартом СТБ ИСО/МЭК 12207–2003. посвящен стратегиям разработки ПС и систем. Проанализированы три базовые стратегии разработки: каскадная, инкрементная, эволюционная. Рассмотрены модели ЖЦ ПС и систем, реализующие данные стратегии, оценены их достоинства, недостатки и области применения. рассмотрены принципы выбора модели ЖЦ ПС и систем, исходя из условий конкретного проекта. Приведена классификация проектов и процедура выбора модели ЖЦ, предложенные Институтом качества программного обеспечения SQI (Software Quality Institute, США). Посвящен рассмотрению классических методологий структурного проектирования ПС. Рассмотрены общие принципы и методы реализации структурного программирования, модульного проектирования, нисходящего проектирования, методов расширения ядра. Данные методологии являются основой ряда современных методологий и технологий

разработки ПС. Описаны такие характеристики структурного разбиения программ на модули, как связность и сцепление.

В книге Андерса Хейлсберга, Язык программирования С# [26] содержится наиболее полная информация по этому языку. Структура книги – сборник спецификаций, в которые включены синтаксис, образцы кода, сопутствующие материалы, а также примеры. В нее включено описание новых возможностей С# 4.0, таких как, динамическое связывание, необязательные и именованные параметры, контравариантные и ковариантные обобщенные типы. В книге, в каждой главе, содержатся комментарии известных мэтров программирования.

В книге Джозефа Албахари С# 6.0. Справочник. Полное описание языка [2] позволяет получить точные ответы практически на любые вопросы по С# 6.0, CLR и основным сборкам .NET Framework. После своего появления в 2000 году С# стал языком с выдающейся гибкостью и мощностью, но его постоянный рост и совершенствование означают необходимость в дополнительном изучении.

В книге Павловской Татьяны Александровны С#. Программирование на языке высокого уровня [14] последовательно изложены следующие ключевые темы: знакомство с платформой .NET, среда разработки Visual Studio, переменные и операторы, классы, работа с массивами и строками, интерфейсы и структурные типы, делегаты, события и потоки выполнения, работа с файлами, сборки, библиотеки, атрибуты, директивы, структуры данных, коллекции и классы-прототипы, и многое другое.

Книга предназначена для тех, кто еще только начинает осваивать С#, независимо от того, новичок ли это, или профессиональный программист, решивший освоить новый язык.

В.А. Ганжа в сотрудничестве с С.Ф. Липницком в своём учебном пособии «Компьютерные сети. Введение»[4] разбирают проблемы построения, работы и обслуживания локально вычислительных компьютерных сетей как необходимого элемента образования ИТ-специалиста. Учебно-методическое

пособие содержит небольшую теоретическую часть в виде слайдов к лекциям и ряд лабораторных работ для практического исполнения, апробированных авторами на протяжении ряда лет на занятиях со студентами.

В легкой и в то же время в познавательной форме передаёт свои мысли Макаровский Д.Д. в своей книге «100 способов ускорить работу вашего компьютера» [9]. В книге собраны лучшие примеры, которые раскроют все возможности техники. Читатель откроет для себя множество полезных возможностей, которые смогут сделать компьютер максимально быстрым и производительным.

Прочитав книгу читающий узнает много увлекательных и интересных возможностей своего компьютера. Научится самостоятельно переустанавливать операционную систему и заменять детали компьютера, при необходимости их замены.

### **1.2.2 Обзор интернет-источников**

В Интернете существует множество ресурсов в свободном доступе. Рассмотрим несколько свободных ресурсов.

Интерактивный учебник по Visual C# находящийся на сайте «Microsoft» [7], содержит обширную базу по работе со средам и информацию по работе с языками. Интерактивный учебник по Visual C# представляет собой серию ознакомительных уроков по среде IDE Visual C#, экспресс-выпуск, в которых содержатся сведения по созданию первого приложения и разработке пользовательского интерфейса приложения.

На сайте «CyberForum» [27] можно найти интересующую информацию по различным языкам программирования, в созданных постах, или задать свой вопрос. На который ответят быстро и понятно. Сайт также хранит информацию по всей области компьютерной техники и электроники.

Материал на электронном ресурсе «Уроки программирования с нуля» [23] содержит информацию для пользователей начинающих изучать C#.

Которая построена в порядке усложнения материала: начиная от простого определения переменной и заканчивая написание собственного приложения.

Сайт «Kbyte.ru Исходные коды. Примеры С#» [8] содержит базу исходных кодов программ написанные опытными программистами и рядовыми пользователями. Данный ресурс будет полезным как начинающим, так и опытным «писателям» строк кода.

В электронном курсе «НОУ Интуит. Основы программирования на С#» [12] представлено полное описание языка С#. Подробно рассматриваются классы, структуры, интерфейсы, делегаты. Особое внимание уделяется наследованию и универсальным классам.

Рассматривается среда разработки Visual Studio .Net и классы библиотеки FCL каркаса Framework .Net. Изложение сопровождается большим числом примеров. В учебном курсе представлено полное описание языка С#.

На сайте «НОУ Интуит. С# для школьников» [13] имеется книга для дистанционного обучения, который обучит юного с объектно-ориентированным программированием на примере реальных программ в среде Windows.

Статья на сайте «Программирование» [19] ознакомит пользователя с программированием. Ответить на вопрос «Что такое программирование?», какие бывают языки для программирования, классификация.

Сайт «Хабрахабр» [25] представляет собой профессиональную интернет площадку, содержащую статьи для большого круга специалистов: программистов, разработчиков, администраторов, тестировщиков, дизайнеров, верстальщиков и других специалистов, работа которых связана с информационными технологиями (ИТ).

Статья на сайте «Пример работы с web-камерой» [17] демонстрирует исходные коды проекта по работе с web-камерой компьютера. Описаны применяемые модули и классы.

Статья на сайте «С#. Берём камеру под контроль» [28] описывает работу web-камеры, как камеры наблюдения и предоставляет листинг и исходные файлы для ознакомления.

Статья на сайте «Захват видео с помощью OpenCV С#» [5] описывает способы применения библиотеки OpenCV для работы с web-камерой.

Статья на сайте «Захват видео при помощи Emgu» [6] описывает способы применения библиотеки Emgu для работы с web-камерой.

Материал, продемонстрированный в статье «Неисправности клавиатуры, мыши, периферийных устройств» [10] описывает неисправности периферийных устройств и способы их устранения.

В статье «Почему перестала работать веб камера? Проблемы с веб камерой» [15] рассматриваются проблемы работы с устройством веб камера.

Статья «Воспроизведение звука в приложении» [16] описывает синтаксис подключения и озвучивания аудиофайлов в приложении.

Статья «Проверка видеокарты на производительность, тест на стабильность» [18] обучить производить тестирование видеокарты на производительность и стабильность оборудования.

Статья «Ремонт жестких дисков компьютеров» [20] описывает проблемы, возникающие при эксплуатации жёсткого диска. Научит производить диагностику и ремонт жёстких дисков.

Сайт «Memtest» [29] содержит подробное описание программ для диагностики памяти и других устройств на материнской плате компьютера.

Статья «Неисправности ПК и устранение их» [11] содержит информацию об основных неисправностях, возникающих при эксплуатации персонального компьютера, и о способах диагностики и ремонта.

Статья «Статистика поломок ноутбуков» [22] приводит статистику, по каким проблемам с ноутбуками обращаются в сервисные центры по ремонту компьютеров и ноутбуков.

Статья «Устранение неисправностей и ремонт ПК своими руками на 100%» [24] представляет из себя выдержку из оригинальной книги разме-



щённой в web форме. В статье описаны часто возникающие проблемы с компьютером, диагностика компонентов и устранение проблем.

Таким образом, проведенный анализ печатной литературы и Интернет-источников показал, что имеется большое количество информации по интересующей области.

### **1.2.3 Вывод по первой главе**

Каждый студент компьютерных профилизаций должен знать как проводить диагностику компьютера. Проводить во время профилактики оборудования, чтоб избежать дальнейших проблем. Имеется множество литературных источников в видео книжных изданиях, научно-популярных журналов и, наверное, самый главный источник знаний – это интернет.

Создание практикума позволить получить первоначальные знания и умения по диагностике компьютера с использование специализированного ПО, но возникает необходимость в разработке дополнительного модуля диагностики периферийного оборудования.

## 2 ОПИСАНИЕ ПРАКТИКУМА И МОДУЛЯ ДИАГНОСТИКИ

### 2.1 Педагогический адрес

Практикум по диагностике компьютера с модулем тестирования периферийного оборудования ориентирован для обучения студентов профиля 44.03.04 Профессиональное обучение (по отраслям) профилизации «Информационные технологии» по дисциплине «Архитектура ПК и периферийные устройства» или «Защита сетевых информационных систем».

Также может быть использован для обучения специалистов сервисных центров и системных администраторов отдела развития информационных сетей и технического сопровождения РГППУ.

Рабочее место мастера по ремонту компьютерной техники показано на рисунке 7.

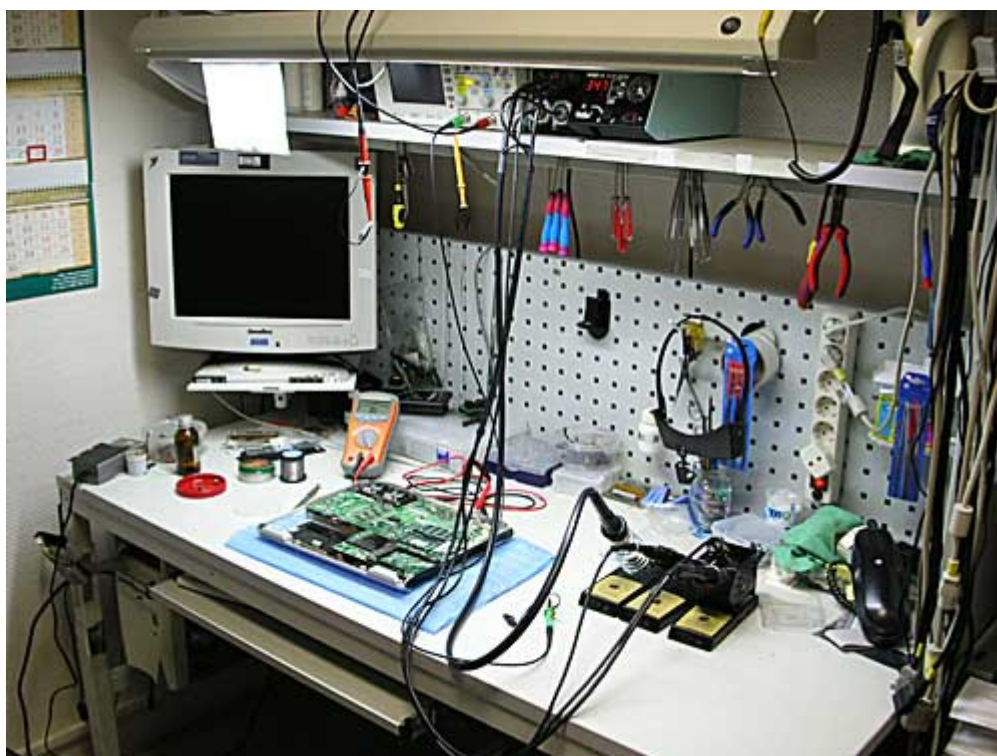


Рисунок 7 – Рабочее место

## 2.2 Общие требования по созданию лабораторных практикумов

Лабораторный практикум является частью теоретической и профессиональной подготовки студентов в Федеральном государственном автономном образовательном учреждении высшего профессионального образования Российского государственного профессионально–педагогического университета.

Значение лабораторного практикума заключается в:

- практическом освоении студентами научно-теоретических положений изучаемой дисциплины;
- овладении техникой экспериментирования соответствующей отрасли науки;
- применении полученных знаний для решения учебно-исследовательских, а затем реальных экспериментальных и практических задач.

Целью выполнения лабораторного практикума является приобретение студентами навыков и умений, необходимых для профессиональной деятельности выпускника.

Дидактическими целями лабораторного практикума являются:

- экспериментальное подтверждение и проверка существующих теоретических положений;
- формирование практических умений и навыков работы с измерительными приборами, аппаратами, компьютерной техникой, лабораторными установками, технологическим оборудованием, составляющих часть профессиональной подготовки;
- формирование исследовательских умений наблюдать, сравнивать, анализировать, устанавливать зависимости, делать выводы, самостоятельно вести исследования, оформлять результаты;
- повышение познавательной активности и самостоятельности работы студентов в ходе выполнения лабораторного практикума путем организации поэтапного контроля их работы;

- усиление практической направленности образовательного процесса;
- углубление теоретических знаний и освоение приемов, методов и способов исследования объектов изучения.

По своему назначению лабораторные практикумы можно классифицировать:

- практикумы вводные или измерительные, которые проводятся по общенаучным и общетехническим дисциплинам. Их цель – проиллюстрировать основные закономерности изучаемой науки, ознакомить студентов с техникой эксперимента, теорией погрешностей и методами обработки экспериментальных данных, с устройством и принципом работы измерительных приборов;
- практикумы, обеспечивающие накопление знаний и практических навыков при усвоении общих курсов и подготовку к изучению специальных дисциплин и методов научных исследований;
- практикумы по специальным дисциплинам и дисциплинам специализации, обеспечивающие практическую подготовку студентов, формирование навыков профессиональных и научных исследований в научной и производственной деятельности.

По характеру выполняемых студентами заданий лабораторные занятия подразделяются на:

- ознакомительные, предпринимаемые с целью закрепления и конкретизации изученного теоретического материала;
- аналитические, ставящие своей целью получение новой информации на основе формализованных методов;
- творческие, связанные с получением новой информации путем самостоятельного выбранных подходов решения задач.

Формами организации студентов на лабораторных работах могут быть:

1. Фронтальная – все студенты выполняют одновременно одну и ту же работу.

2. Групповая – одна и та же работа выполняется подгруппами по 25 человек.

3. Индивидуальная – каждый студент выполняет индивидуальное задание.

Лабораторный практикум разрабатывается на базе учебно–лабораторного оборудования кафедры, средств компьютерной поддержки, программных моделей изучаемых процессов и явлений.

Особого подхода требует технология проектирования и проведения лабораторного практикума с использованием web-технологий.

Компьютерные технологии получения, хранения и преобразования информации при развитых интерфейсных системах ввода–вывода позволяет обеспечить проведение лабораторных практикумов более эффективным. Задача практикума в этом случае не просто научить студента «нажимать кнопки» по определенным правилам, но обеспечить формирование необходимых профессиональных умений, экономя время и позволяя провести опыты, которые трудно или невозможно выполнить стандартными приемами.

Программное и информационное обеспечение лабораторного практикума, особенно в части программного интерфейса, должно корректно функционировать и обладать интуитивно понятным, дружелюбным интерфейсом.

Средства мультимедиа позволяют представить учебный материал в увлекательной, динамичной форме, а инженерные конструкции, устройства, элементы – как движущиеся трехмерные объекты, тем самым в полной мере раскрывая их конструкцию и принцип действия.

Современные специализированные программные пакеты ориентированы облегчить труд разработчика при работе с большим количеством материала в процессе создания электронного мультимедийного продукта, а также при дополнении материала в уже готовом продукте.

Это позволяет создавать хорошие электронные продукты с точки зрения психологии восприятия в условиях компьютерного обучения, а также в плане минимизации трудовых, временных, финансовых затрат.

В то же время, появляется и список требований, предъявляемых к таким лабораторным практикумам. Помимо требований к оформлению текста, появляются требования к оформлению мультимедийной информации содержимого.

Рассмотрим список типовых требований более детально.

**Достоверность информации.** Это требование является наиболее актуальным, поскольку количество информации, содержащейся в Интернете, растёт в геометрической прогрессии, но при этом, сколько источников, столько и мнений. Весьма сложно становится находить именно достоверную информацию о интересующем направлении. Именно поэтому, при составлении лабораторного, стоит использовать только достоверные источники материала.

**Наглядность представления.** Как ни странно, но чем внешне привлекательнее электронный продукт, тем больший круг пользователей он получает. Много зависит от оформления электронного продукта, будь то фон для текстовых блоков или же часть фрагмент с представлением, скажем, видеоролика. Первую оценку от читателя мультимедийный продукт получает за внешний вид.

**Удобство использования.** Так как лабораторный практикум, как правило, состоит из большого количества страниц, то самое главное на этапе подготовки правильно произвести структуризацию материала, чтобы читатель мог без лишнего труда найти необходимую ему информацию, при этом не пролистывая страницы и разделы, которые ему конкретно в данный момент не нужны. Для этого необходимо обеспечить максимальное удобство навигации по страницам электронного продукта.

## **2.3 Выбор средств реализации практикума и модуля**

В результате проведённой работы было создано электронное учебное пособие с применением следующих технологий:

1. HTML (язык разметки гипертекста) – стандартный язык разметки гипертекста использовался для верстки html-страниц лабораторного практикума.
2. CSS (каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки;
3. C#(Си шарп) - объектно-ориентированный язык программирования;
4. Microsoft Visual Studio 2010 C# - среда для разработок от компании Microsoft.

При разработке учебного продукта выбор был сделан в пользу данных web-технологий, так как данное решение является кроссплатформенным и даёт возможность изучать лабораторный практикум под любой операционной системой.

Для создания рисунков пособия применялся Adobe Photoshop CS6 – мощный графический редактор от компании Adobe, заслуженно является лучшим графическим редактором в мире, для обработки фотографий и растровых рисунков. При работе с лабораторным практикумом применялся для создания и редактирования, вставочных изображений в практикум.

## **2.4 Интерфейс, структура и навигация практикума**

Для начала работы с учебным пособием необходимо запустить «Практикум.exe» После запуска программы открывается главное диалоговое окно (см. рисунок 8).

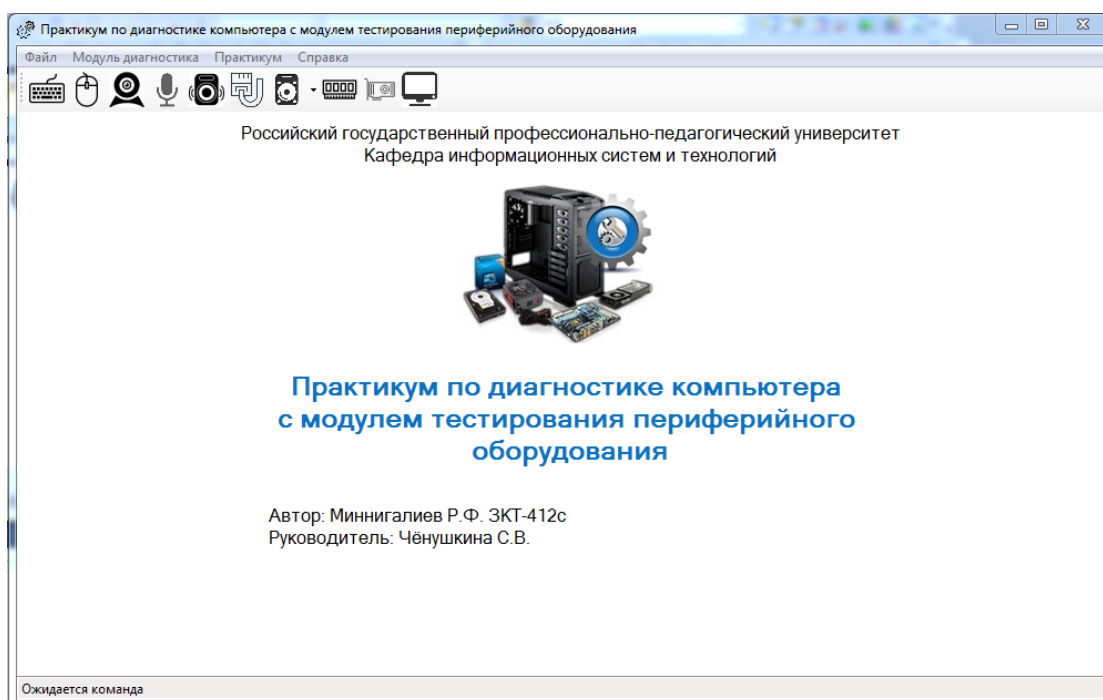


Рисунок 8 – Главное окно практикума

Интерфейс практикума разработан на языке C# и скомпилирован в среде разработок Microsoft Visual C# 2010.

Навигация по практикуму осуществляется с помощью верхнего меню, позволяющая переключается между лабораторными работами и модулем диагностики, из которого запускаются модули диагностики и для удобства модули продублированы на панель быстрого доступа (см. рисунок 9 и рисунок 10).

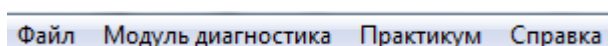


Рисунок 9 – Главное меню программы

Изображения, вставленные на панели, обозначает запуск модуля по диагностике соответствующего компонента, или периферийного оборудования персонального компьютера.



Рисунок 10 – Панель быстрого доступа

Подробная структура лабораторного практикума отражена на рисунке 11.



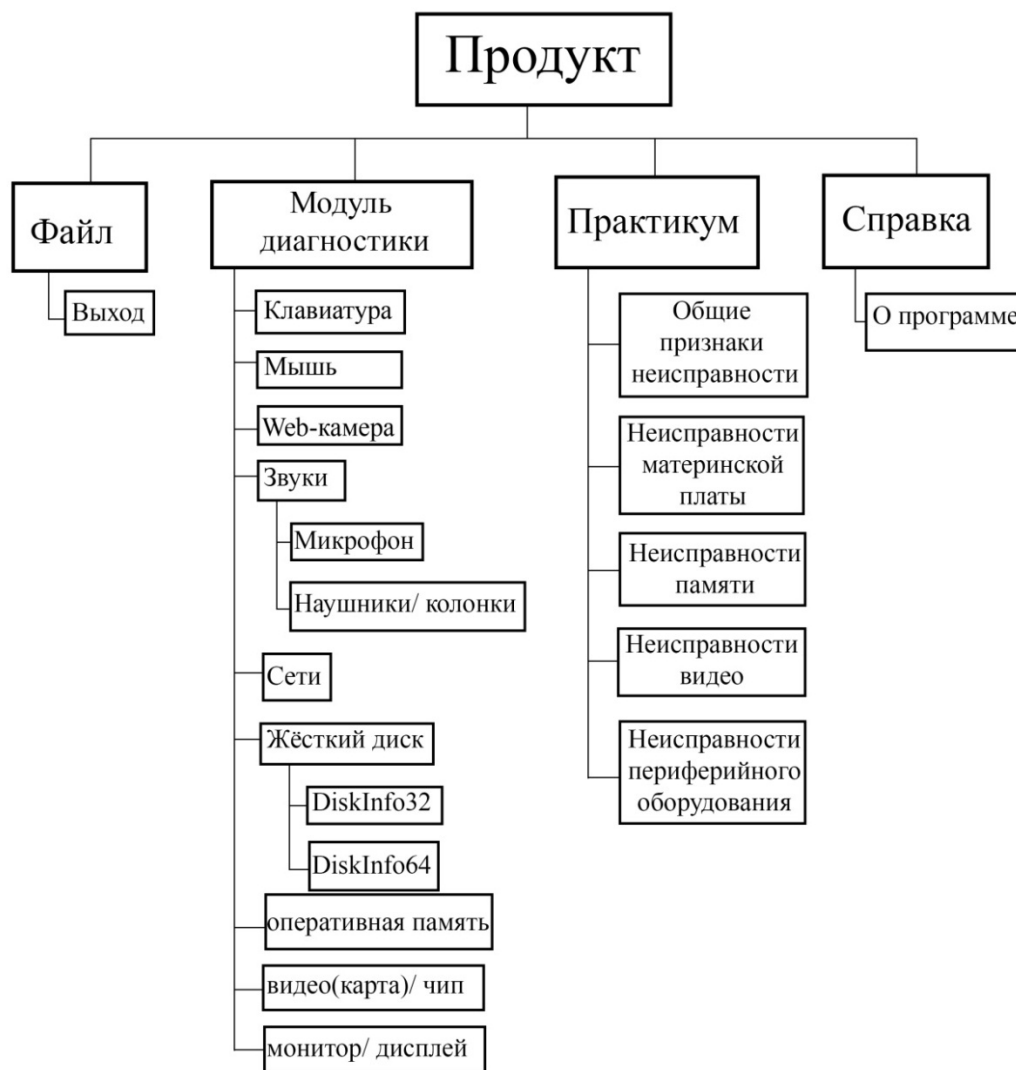


Рисунок 11 – Структура лабораторного практикума

Для просмотра лабораторных работ практикума используется web-браузер, оснащённый дополнительными кнопками навигации, слева направо: возврат, обновление и закрытие. Кнопка «назад» возвращается на страницу активной лабораторной работы, если вышли за её пределы, кнопка «обновление» загружает страницу заново, кнопка «закрытие» завершает работу со страницей лабораторной работы и возвращается на главное окно. Дополнительные кнопки показаны на рисунке 12.



Рисунок 12 – Дополнительные кнопки навигации

## 2.5 Описание собственного модуля диагностики оборудования

Основной проблемой разработок это выбор среды для компиляции и языка программирования. В нынешнее время компиляторы поддерживают синтаксисы различных языков программирования, но данный лабораторный практикум писался в среде разработок от компании Microsoft под названием «Microsoft Visual Studio 2010 C#. Как видно из названия писался практикум на объектно-ориентированном языке программирования C#.

Данная версия студии ориентирована только на C#. Студия имеет интуитивно понятный интерфейс, имеется система «Intellisense» это система помогающая выводить слово по первым символам, что удобно не приходится вводить функцию целиком вручную.

Проект разрабатывался под dot net framework 4.0(.Net 4.0). К проекту подключены следующие библиотеки:

- System;
- System.Core;
- System.Data;
- System.Data.DataSetExtensions;
- System.Deployment;
- System.Drawing;
- System.Windows.Forms;
- Microsoft.CSharp;
- NAudio;
- Emgu.CV;
- Emgu.CV.UI;
- Emgu.Util.

Кроме последних 4 библиотек остальные добавляются автоматически при создании проекта на Windows Form.

Библиотека «NAudio» предназначена для работы со звуком, в проекте она применялась для модулей по диагностике микрофона и колонок, библиотека имеет собственные элементы управления.

Библиотеки под управлением «Emgu» предназначены для работы с камерами и с видео. При помощи библиотеки можно записывать видео с web-камер и другие действия с видео, часть функции, которой было применено в практикуме. Модуль «web-камера» подключён практикуму как отдельное приложение, но изначально оно было встроено в единый проект разработанного лабораторного практикума. Данная библиотека отказывалась работать с модулем «клавиатура», которое было разработано первым, в отдельном проекте «Emgu» работала без ошибок, в связи с этим было принято решение о подключении к основному проекту как готовое приложение.

Полный листинг программы содержится в приложении 2.

Обозреватель решений проекта показано на рисунке 13.

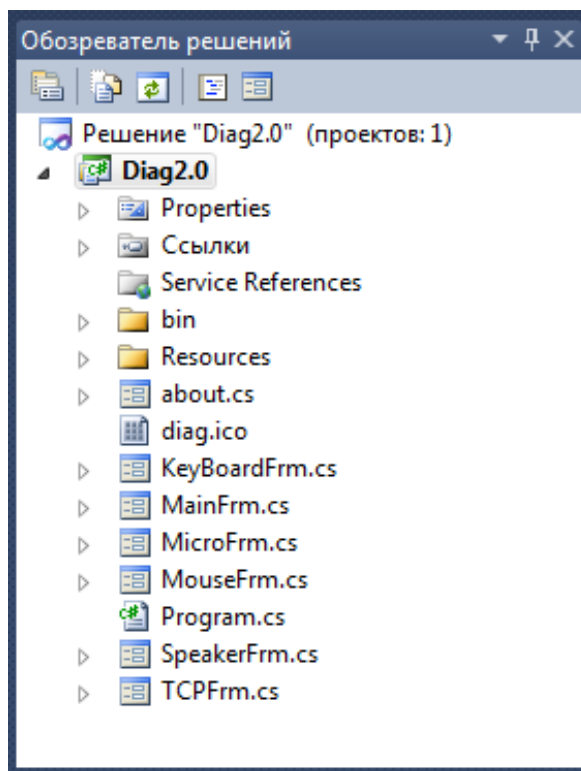


Рисунок 13 – Обозреватель решений

Главное меню продемонстрировано на рисунке 14.

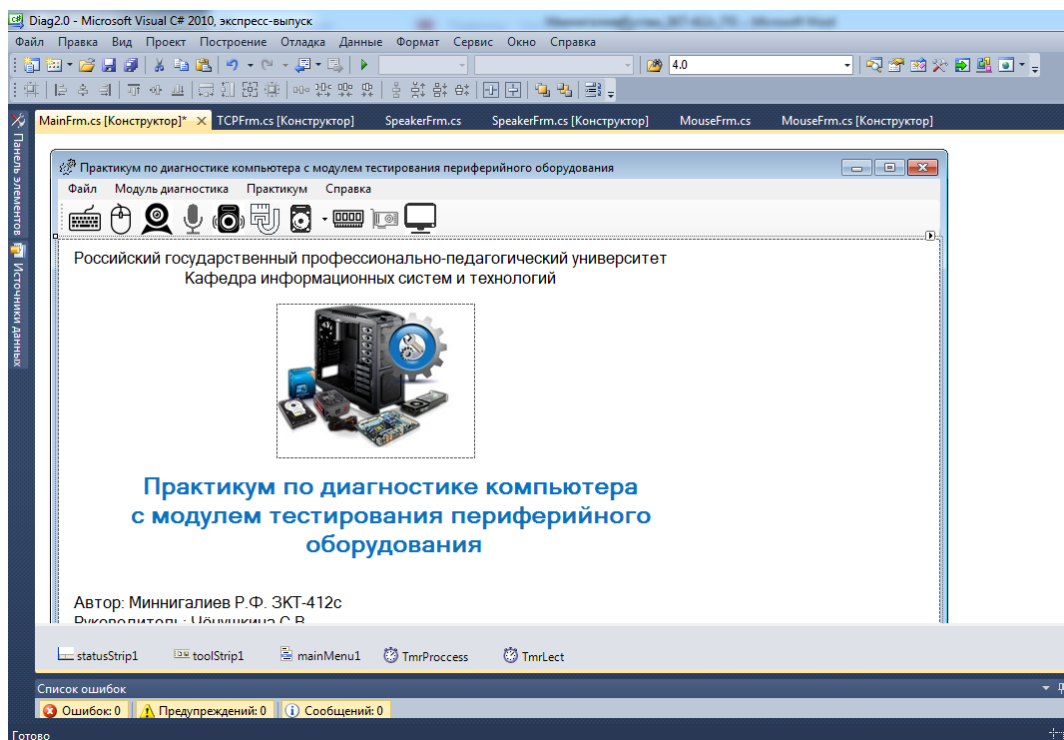


Рисунок 14 – Главное окно проекта

Окно модуля сети показано на рисунке 15.

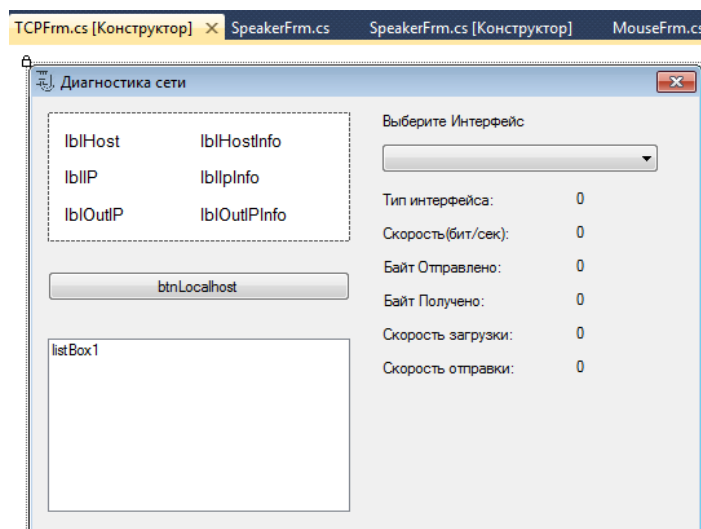


Рисунок 15 – Окно модуля сети

Раздел «Модуль диагностика» состоит 10 пунктов запускающих модули диагностики. Название олицетворяет компонент, который диагностирует модуль. Модули показаны на рисунке 16.

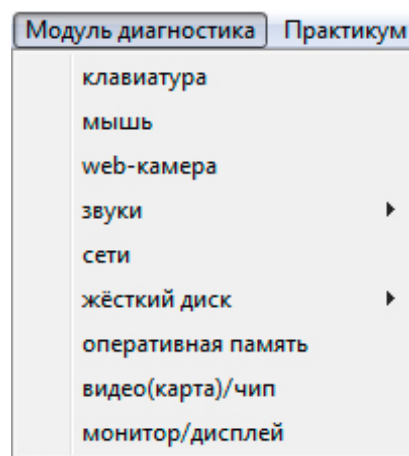


Рисунок 16 – Модуль диагностика

Пункт «звуки» содержит 2 модуля: микрофон и наушники/ колонки (см. рисунок 17).

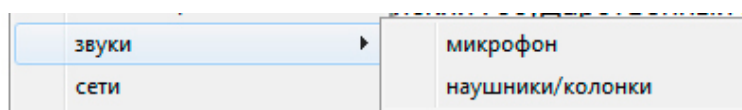


Рисунок 17 – Пункт «звуки»

Пункт «жёсткий диск» предлагает на выбор по архитектуре операционной системы версии для 32 битных систем и для 64 битных систем (см. рисунок 18).

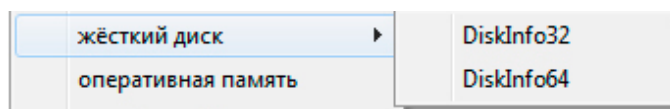


Рисунок 18 – Модуль «жёсткий диск»

Рассмотрим подробно модули диагностики.

**Модуль «клавиатура»** предназначена для проверки работоспособности клавиш клавиатуры и имеет аналогичное расположение клавиш (см. рисунок 19).

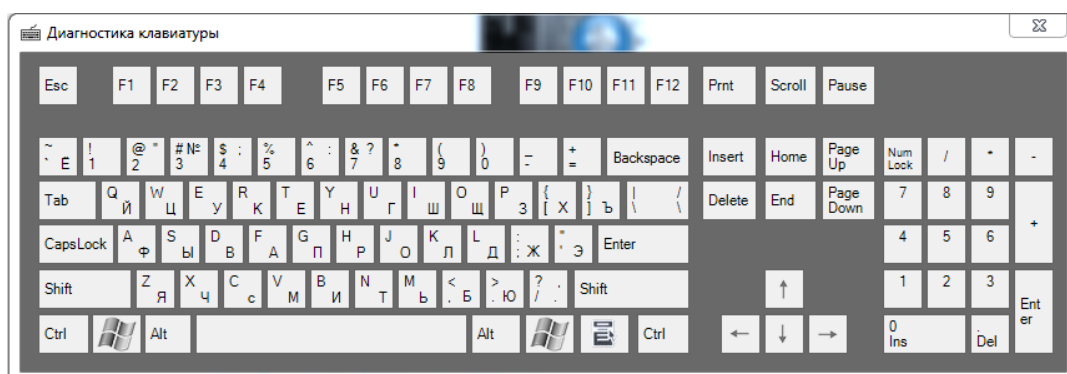


Рисунок 19 – Главное окно модуля клавиатура

Для проверки клавиши достаточно нажать на неё. При нажатии, соответствующий клавише элемент окрасится в красный цвет (применяется цвет Tomato), а при отпуске клавиши цвет станет бледным (применяется цвет Darksalmon), что значит, клавиша была прожата и отпущена (см. рисунок 20 и рисунок 21).

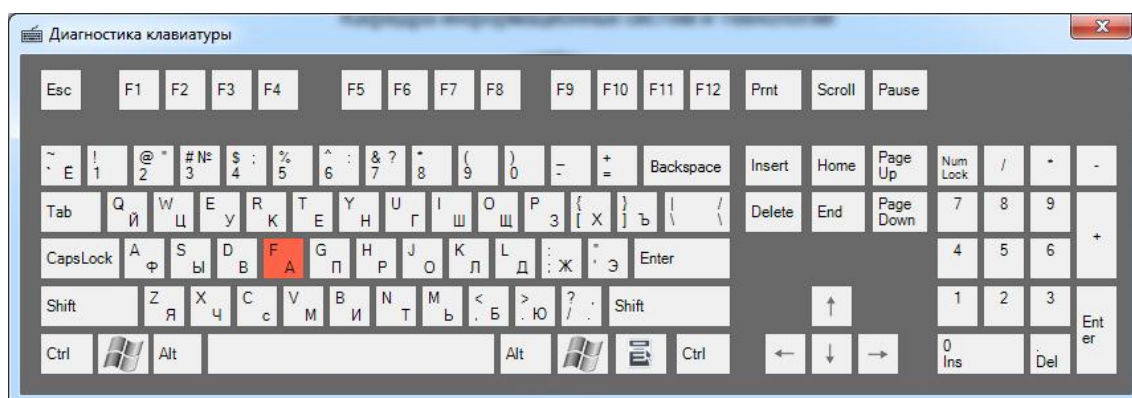


Рисунок 20 – Клавиша нажата



Рисунок 21 – Клавиша отпущена

Если все клавиши клавиатуры были прожаты и не осталось серых кнопок в диалоговом окне, то считается с клавиатурой всё хорошо (см. рисунок 22).



Рисунок 22 – Нажаты все клавиши

В противном случае, нужно вытащить не ответившую клавишу и самостоятельно прожать пластину, на которую давит клавиша при нажатии, если результат остался неизменным то, скорее всего, повреждена дорожка клавиши, ибо сигнал не доходит до компьютера.

**Модуль «мышь».** Модуль предназначен для проверки мышки на нажатие клавиш, скорость двойного клика, отслеживание положения мыши на плоскости. На белом полотне при нажатии левой кнопки мыши будет рисоваться квадратики, при нажатии правой, он перестанет рисовать, размер и цвет которых изменяются в соответствующих полях слева (см. рисунок 23 и рисунок 24).

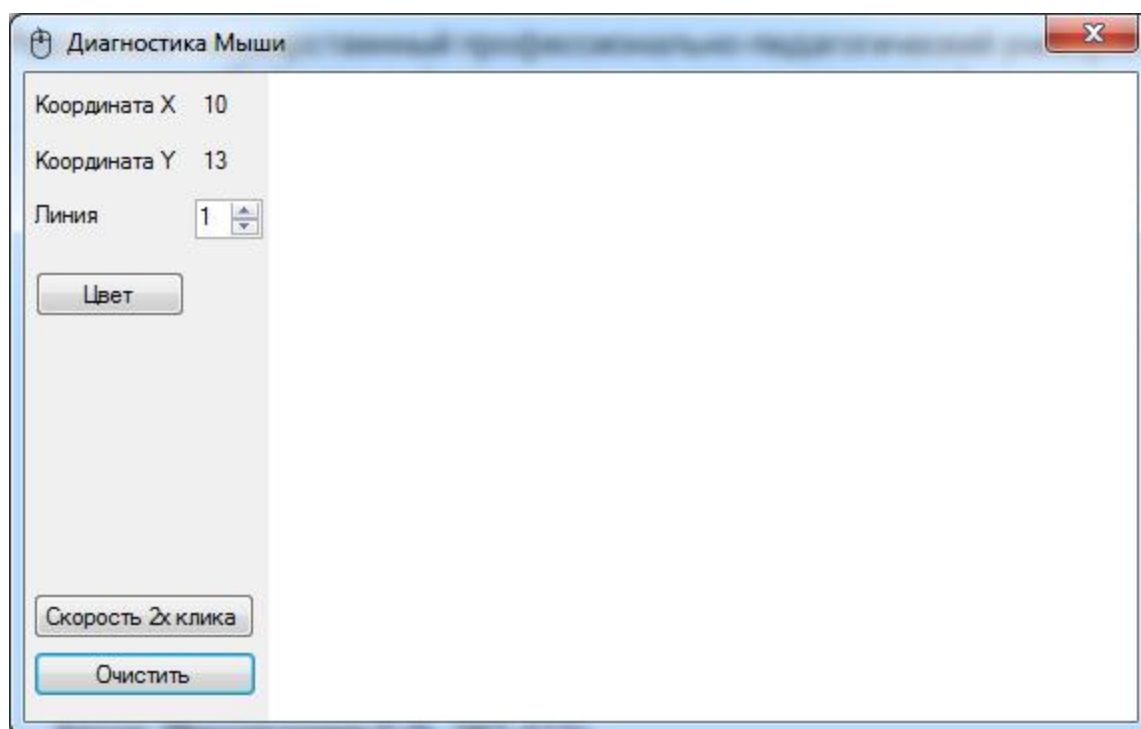


Рисунок 23 – Окно модуля «мышь»

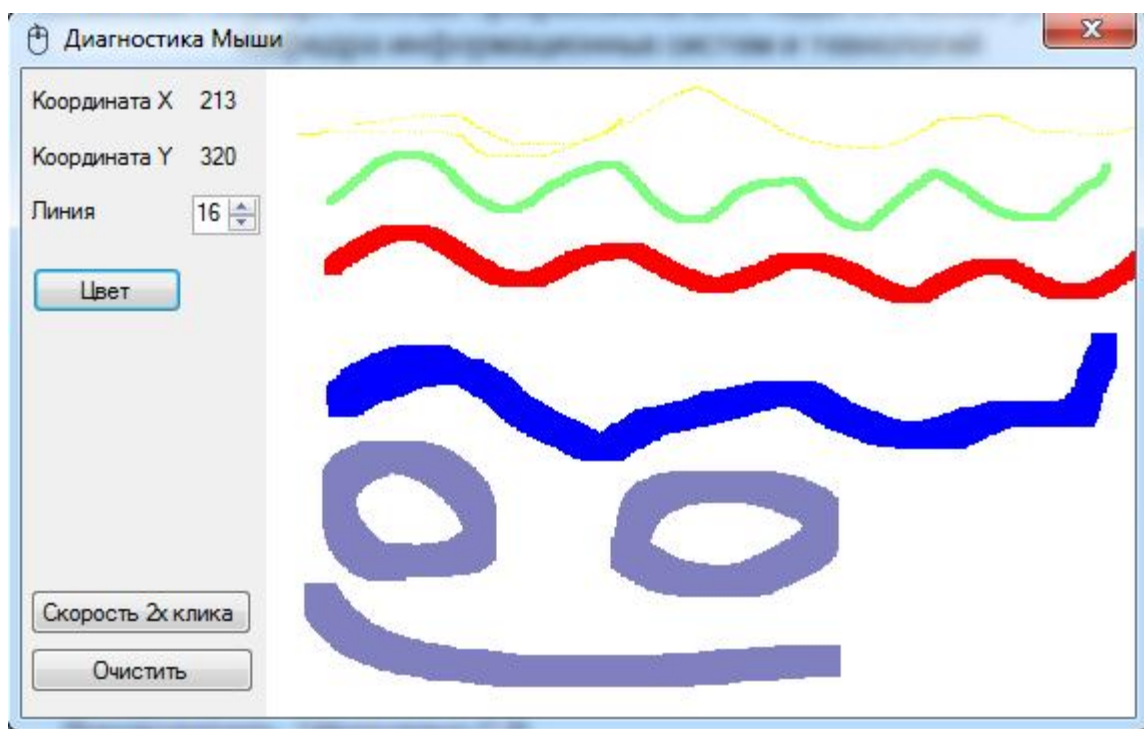


Рисунок 24 – Рисование на полотне

По нажатию кнопки «очистить» полотно станет чистым.

По нажатию кнопки «Скорость 2х клика» отобразится небольшое пространство над кнопкой, на котором необходимо дважды прожать кнопку мыши, но со скоростью достаточной, чтоб оно считалось двойным (см. рисунок 25).

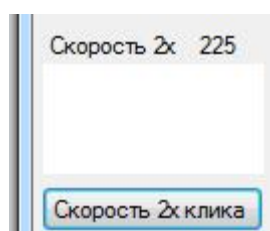


Рисунок 25 – Скорость двойного клика мыши

**Модуль «web-камера».** В «модуле диагностики» пункт «web-камера» разработан быстрый способ проверки работоспособности web-камеры. При благоприятных условиях включится демонстрация изображения с web-камеры, иначе если устройство не было обнаружено программа выдаст предупреждающее сообщение (см. рисунок 26 и рисунок 27).





Рисунок 26 – Web-камера работает

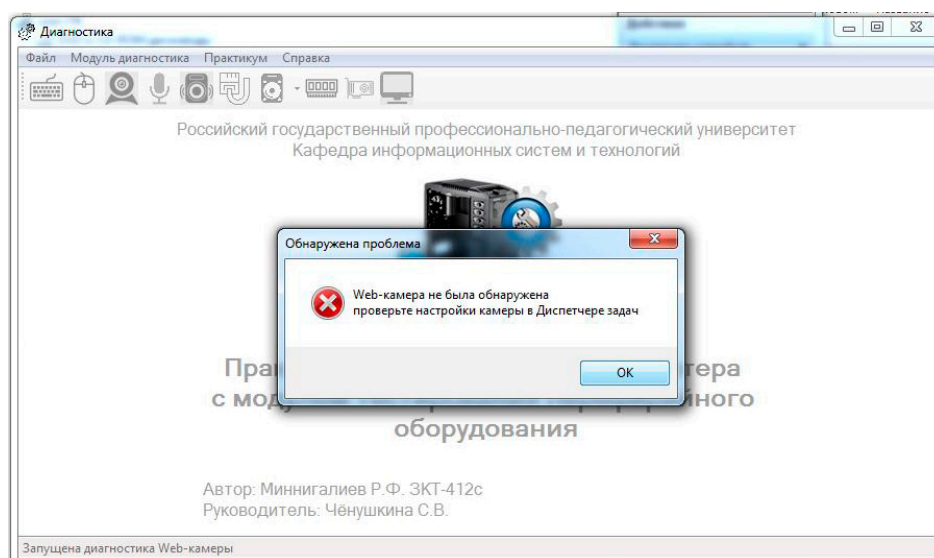


Рисунок 27 – Web-камера не работает

**Модуль «наушники/колонки».** Данный модуль проверяет работоспособность двухканальных колонок, или наушников, путем проигрывания звуковой дорожки для левых наушников – левый канал и отдельно для правых - правый канал. Окно модуля показано на рисунке 28.

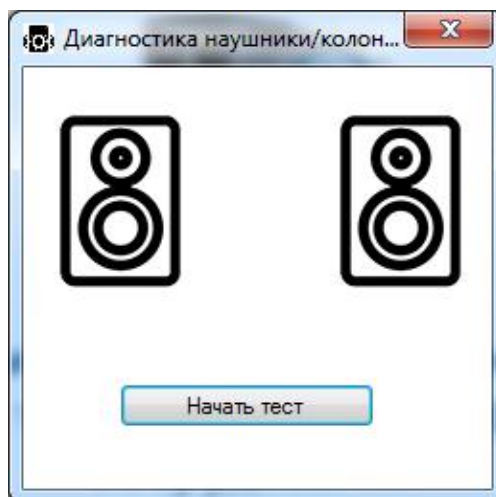


Рисунок 28 – Окно модуля

После нажатия «Начать тест» проигрывается музыкальная мелодия для левого и правого колонок/каналов, на рисунке 29 продемонстрировано отображение при проверке левого канала, на рисунке 30 правого.

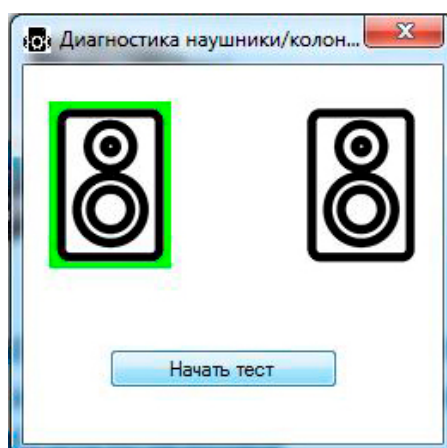


Рисунок 29 – Проверка левого канала

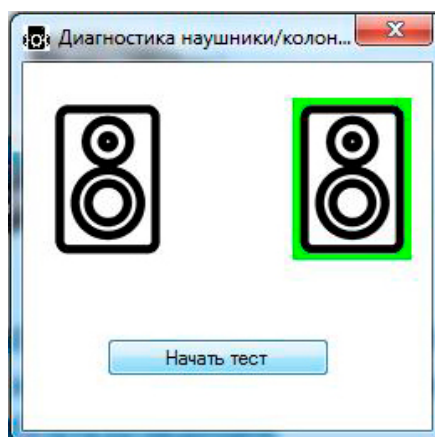


Рисунок 30 – Проверка правого канала

**Модуль «микрофон».** Модуль микрофон проверяет микрофон, выставленный по умолчанию, на приём звуковых сигналов, о чём будет показывать уровень на окне, продемонстрировано на рисунке 31 и на рисунке 32.

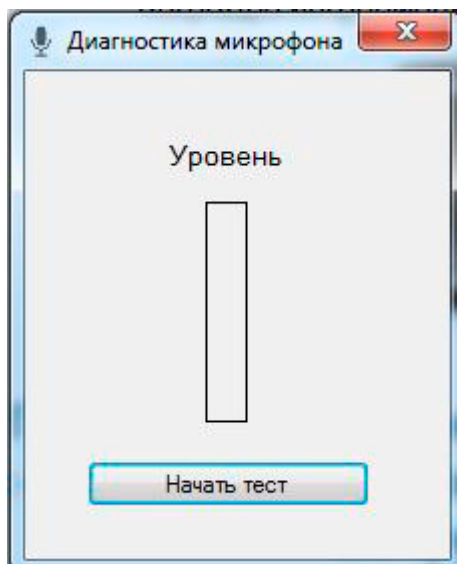


Рисунок 31 – Окно модуля «микрофон»

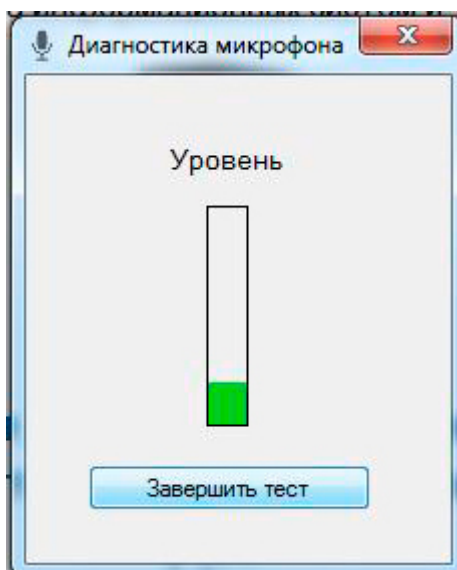


Рисунок 32 – Пример работы модуля

**Модуль «сети».** В окне отображается текущее сетевое имя, локальный IP адрес, глобальный IP адрес(внешний, под которым отображается компьютер в сети интернет), в поле ниже при нажатии кнопки «Вывести список домашней сети» выведутся сетевые имена компьютеров, входящих в один и тот же домен. Справа в выпадающем списке можно выбрать имена сетевых интерфейсов. После выбора отобразится тип интерфейса, пропускная ско-

рость (бит/сек), объём отправленных и полученных данных, текущие скорость загрузки и отправки (в КБ/сек) (см. рисунок 33).

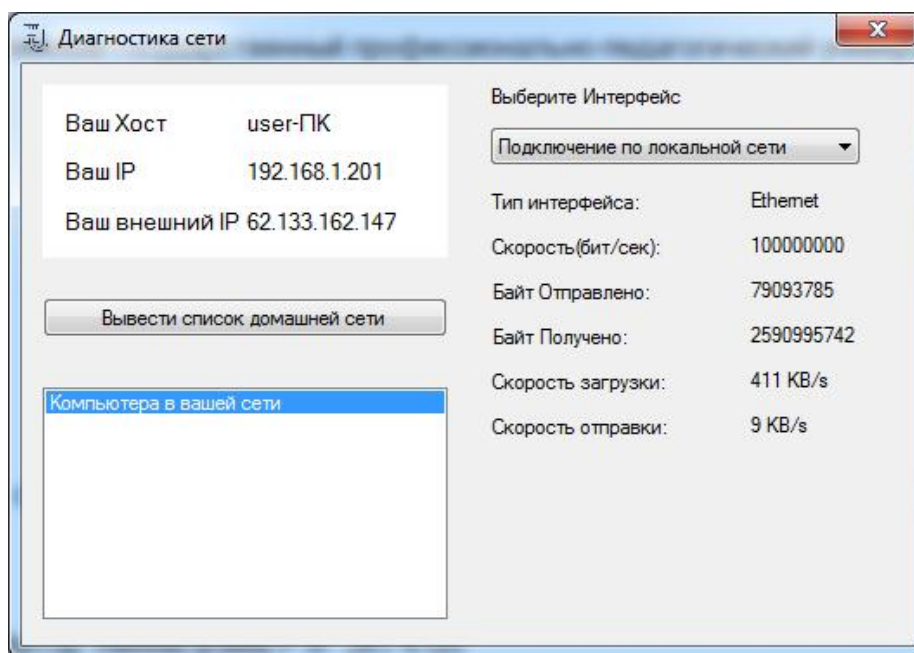


Рисунок 33 – Модуль «сети»

## 2.6 Описание модулей работы с программами диагностики

**Модуль «жёсткий диск».** Модуль предназначен для запуска диагностирующего приложения – CrystalDiskInfo, на поле «жёсткий диск» будет предложен выбор версии приложения по разрядности системы.

Программа CrystalDiskInfo предназначена для проверки носителей информации на S.M.A.R.T. атрибутов, изменения специальных настроек, таких как: управление питанием, регулировка акустического режима и др. Предусмотрен вывод значения температуры накопителя в панель задач.

Программа поддерживает работу с дисками, подключёнными как по SATA разъему, так и по USB, показывая всю его информацию начиная серийным номером диска, версии прошивки и заканчивая общим времени работы.

При включении программа проверить состояние жёсткого диска и вынесет полученные данные в таблицу на главное окно программы. Главное окно приложения, показано на рисунке 34.

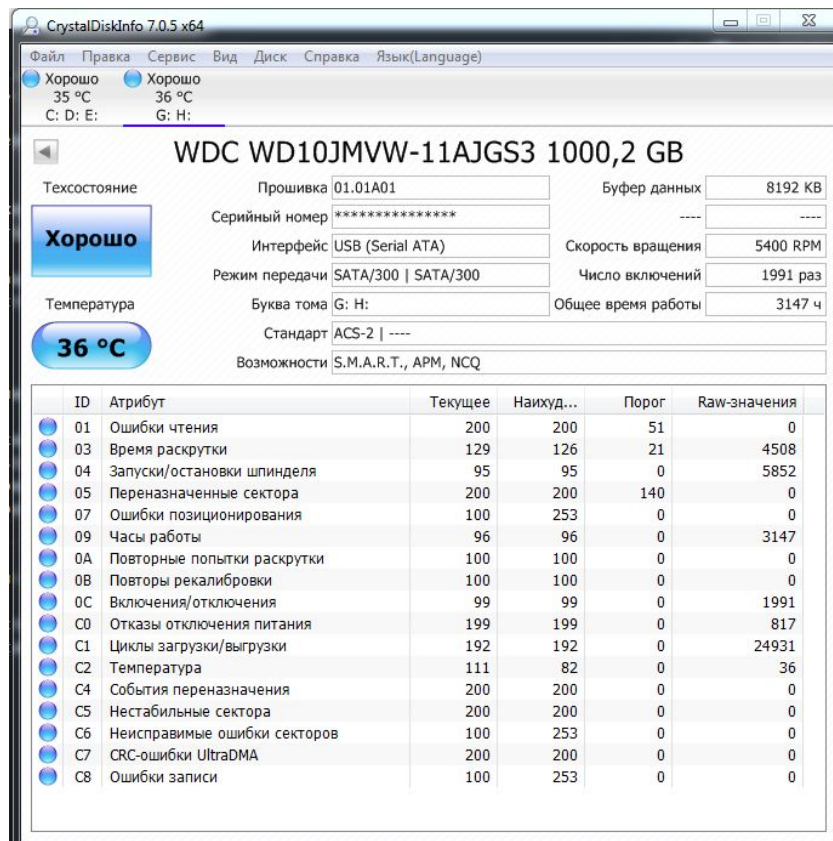


Рисунок 34 – Главное окно CrystalDiskInfo

В настройках программы возможна демонстрация атрибутов SMART в виде графика.

**Модуль «память».** Модуль запускает windows mini-версию популярной программы по проверке памяти Memtest (см. рисунок 35).

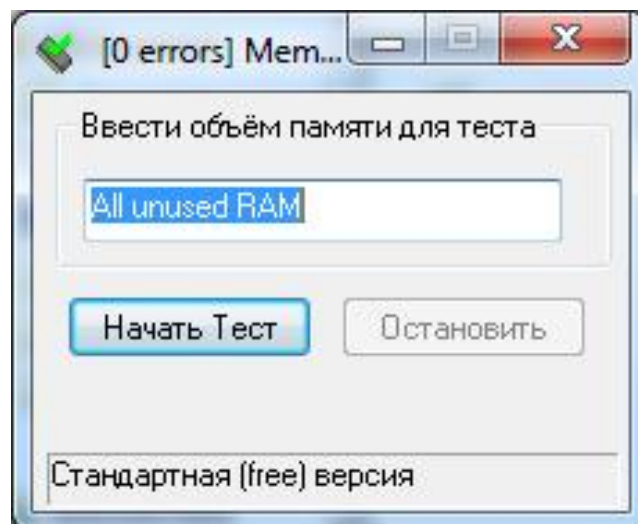


Рисунок 35 – Окно Memtest



Для теста достаточно нажать на кнопку «Начать тест». Процесс тестирования отобразится в нижней части программы. Программа отобразит количество ошибок после завершения теста.

**Модуль «видео (карта)/чип».** Модуль запускает программу для тестирования видео ядра «FurMark». Главное окно программы продемонстрировано на рисунке 36.

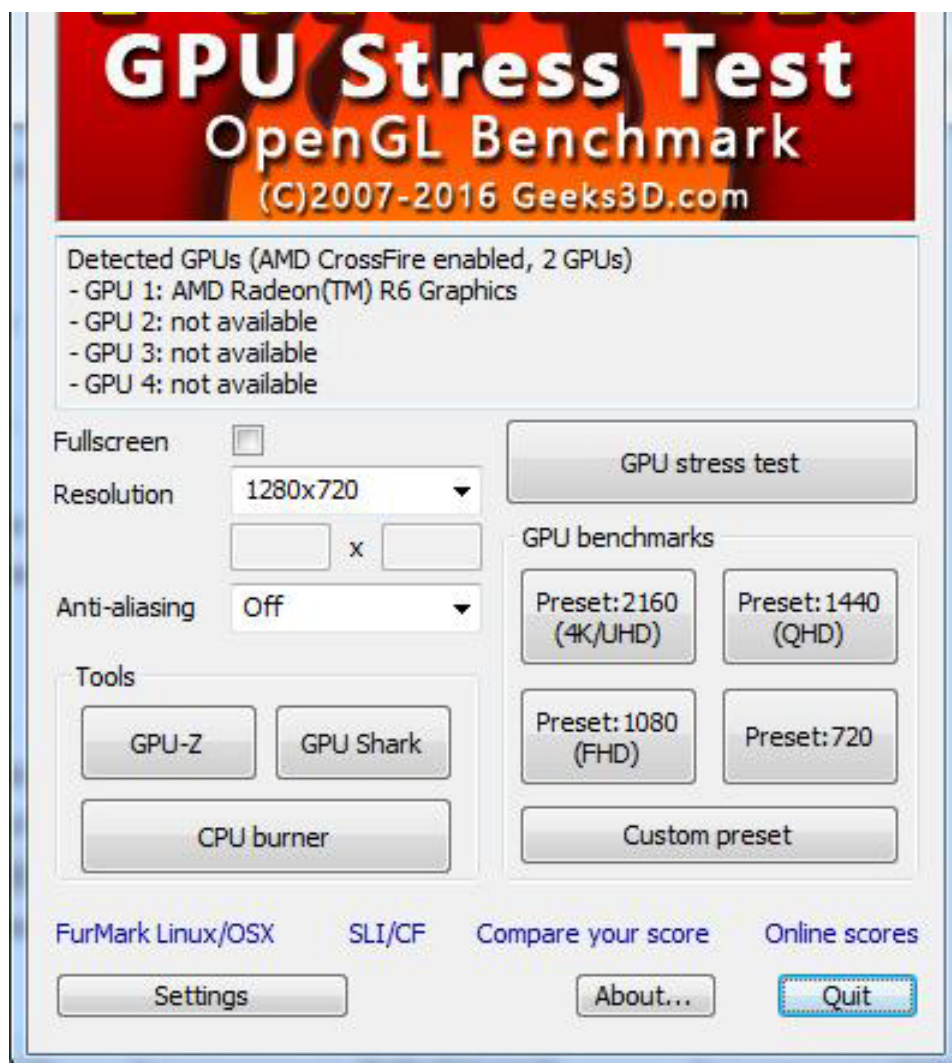


Рисунок 36 – окно «FurMark»

Так же «FurMark» имеет встроенное приложение GPU-Z, которая показывает подробную информацию о видеокарте или видеочипе (в случае с ноутбуком). Приложение показывает название видеомодуля, его объём, технические характеристики и другие параметры, показано на рисунок 37.

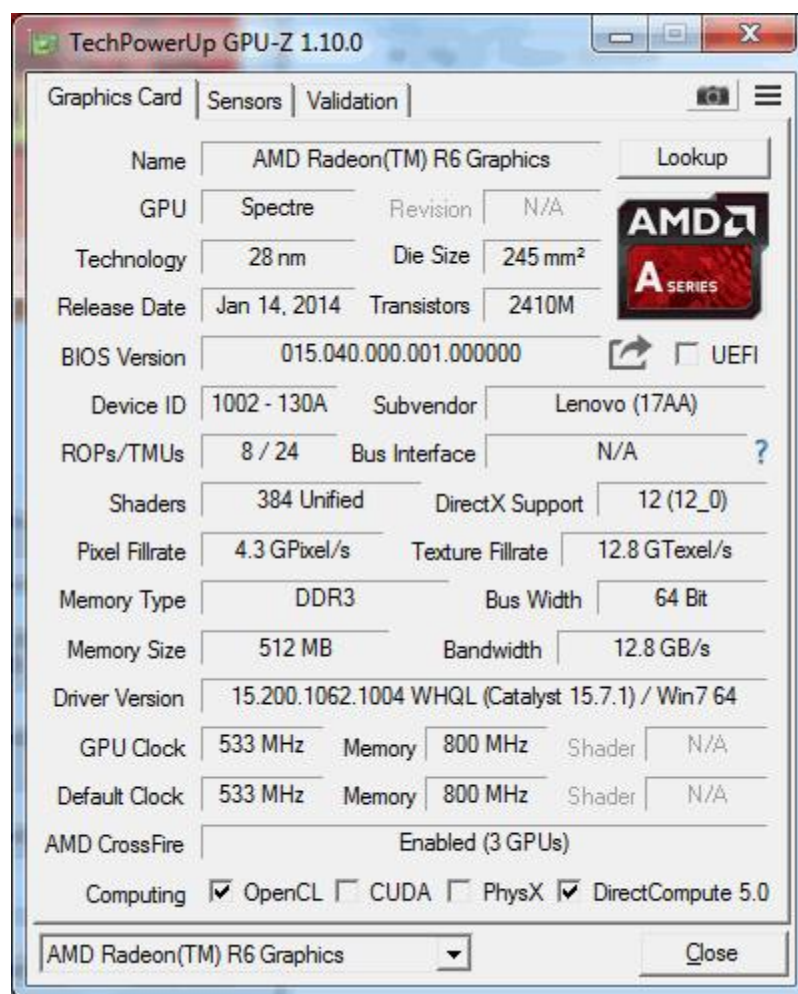


Рисунок 37 – Окно GPU-Z

Перед началом теста необходимо выбрать режим в котором будет происходить тест: в оконном или в полноэкранном. Достигается выбором строки «Fullscreen».

Второе что следует выбрать – в поле «Resolution» разрешения экрана, и далее выбрать типа теста:

1. Бесконечный Стресс тест;
2. Четыре из предложенных разрешениях:
  - разрешение 2160(4K/ UltraHD);
  - разрешение 1440(QHD);
  - разрешение 1080(FullHD);
  - разрешение 720 (HD).
3. Простой тест без фильтров.

Наглядное отображение порядка действий в программе показано на рисунке 38.

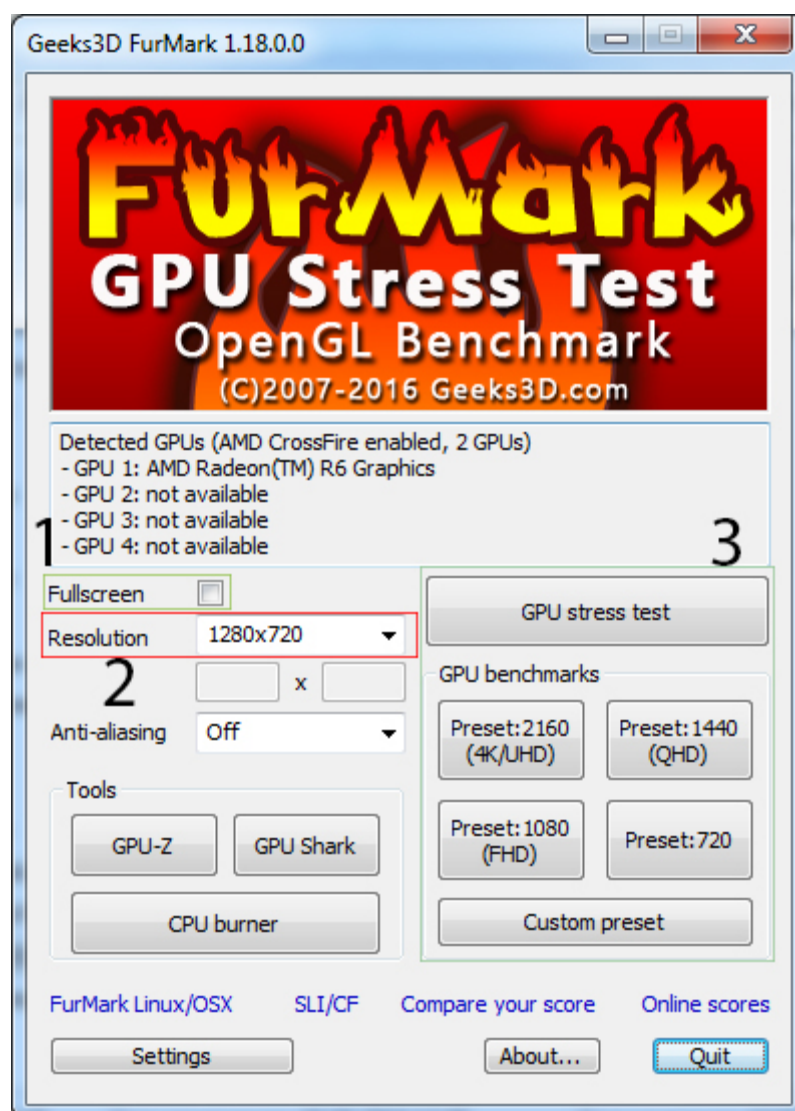


Рисунок 38 – Порядок действий

После проведения подготовительных действий, и после выбора типа теста откроется окно теста, где отображается текущая нагрузка на видео, её температура (минимальная, максимальная и среднее) и пропускная способность кадров в секунду времени(FPS). Окно тестирования видео ядра показано на рисунке 39.





Рисунок 39 – Процесс тестирования

**Модуль «монитор/дисплей».** Модуль запускает программу «TFT монитор».

На главном окне отображается (см. рисунок 40):

1. Название видеокарты.
2. Название используемого монитора.
3. Текущее разрешение экрана.
4. Версия установленной операционной систем.

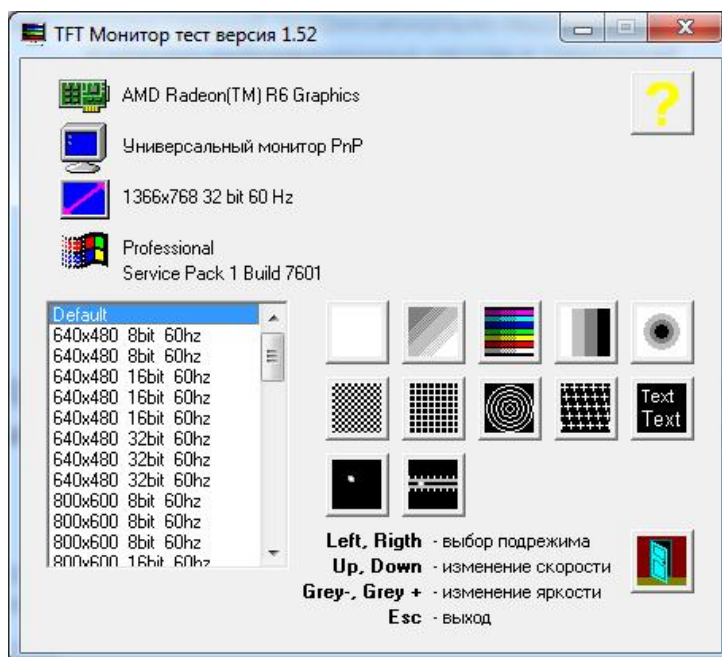


Рисунок 40 – Окно «TFT монитор»

Тестирование на битые пиксели:

1. Из списка необходимо выбрать разрешения экрана с битностью, можно оставить «Default» вариант.
2. Для начала теста, выбирается одна из предложенных тестов справа (при наведении выводится название теста), внизу под тестами есть краткое руководство по изменению некоторых параметров при тесте.

## **2.7 Описание лабораторных работ**

В разделе «Практикум» содержатся 5 лабораторных работ:

1. Общие признаки неисправностей.
2. Неисправности материнской платы
3. Неисправности памяти.
4. Неисправности видео.
5. Неисправности периферийного оборудования.

Лабораторные работы выполняются последовательно, с первой темы. Перед выполнением заданий по лабораторной работе следует ознакомиться с краткой теоритической частью, где описываются методы работы с «модулем диагностика».

По завершению лабораторных работ под номерами с 2 по 5 предлагается самостоятельно заполнить отчёты и сделать вывод о проделанной работе.

По итогу завершения всех лабораторных работ предусматривается тестирование.

### **Лабораторная работа №1 « Общие признаки неисправностей »**

**Тема:** Общие признаки неисправностей.

**Цель:** Научиться производить диагностику персонального компьютера.

**Задачи:**

1. Научиться распознавать признаки неисправностей.
2. Научиться исправлять простые неисправности по уровню сложности решения.

3. Научиться по звуку, издаваемому спикером материнской платы, определять мест возникновения ошибка.

В первой лабораторной работе обучаемый знакомится с распространёнными признаками неисправностей, возникающими во время эксплуатации компьютера, и методами их диагностирования, или исправления, в «домашних условиях». Знакомит с таблицей звуков, издаваемых спикером материнской платы, и их текстовым значением.

Фрагменты лабораторной работы №1 продемонстрированы на рисунке 41, рисунке 42 и на рисунке 43.

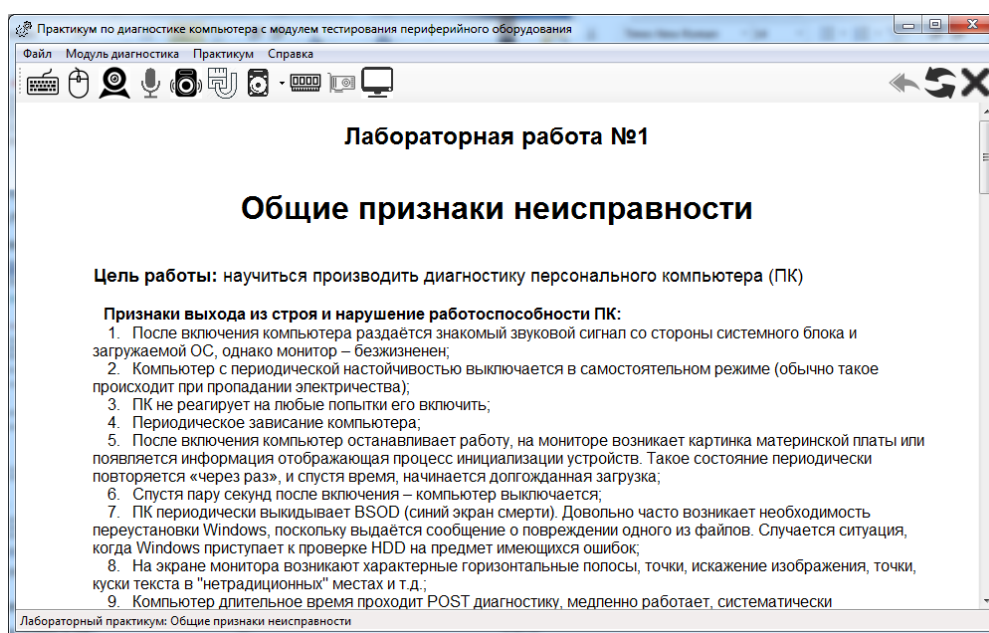


Рисунок 41 – Фрагмент №1 лабораторной работы

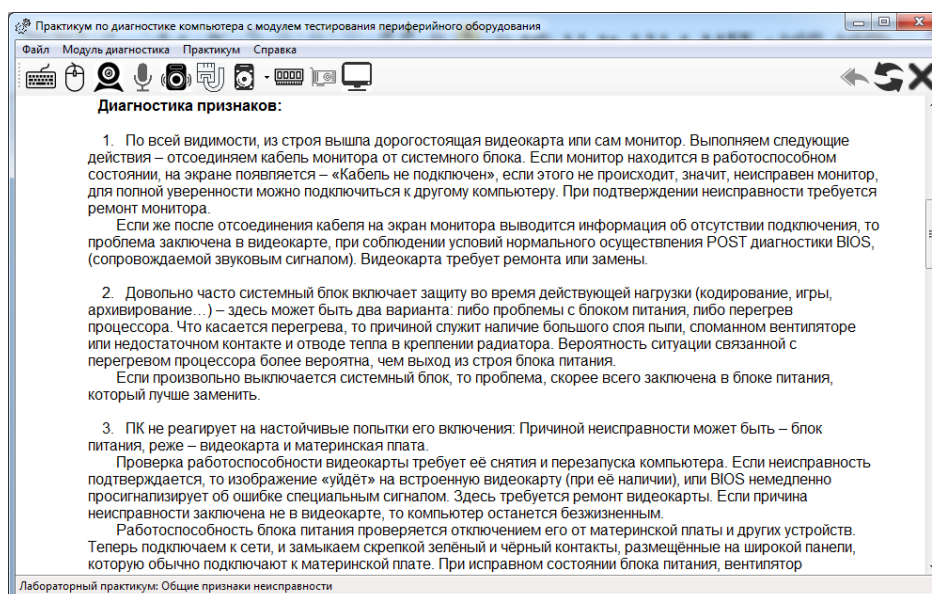


Рисунок 42 – Фрагмент №2 лабораторной работы

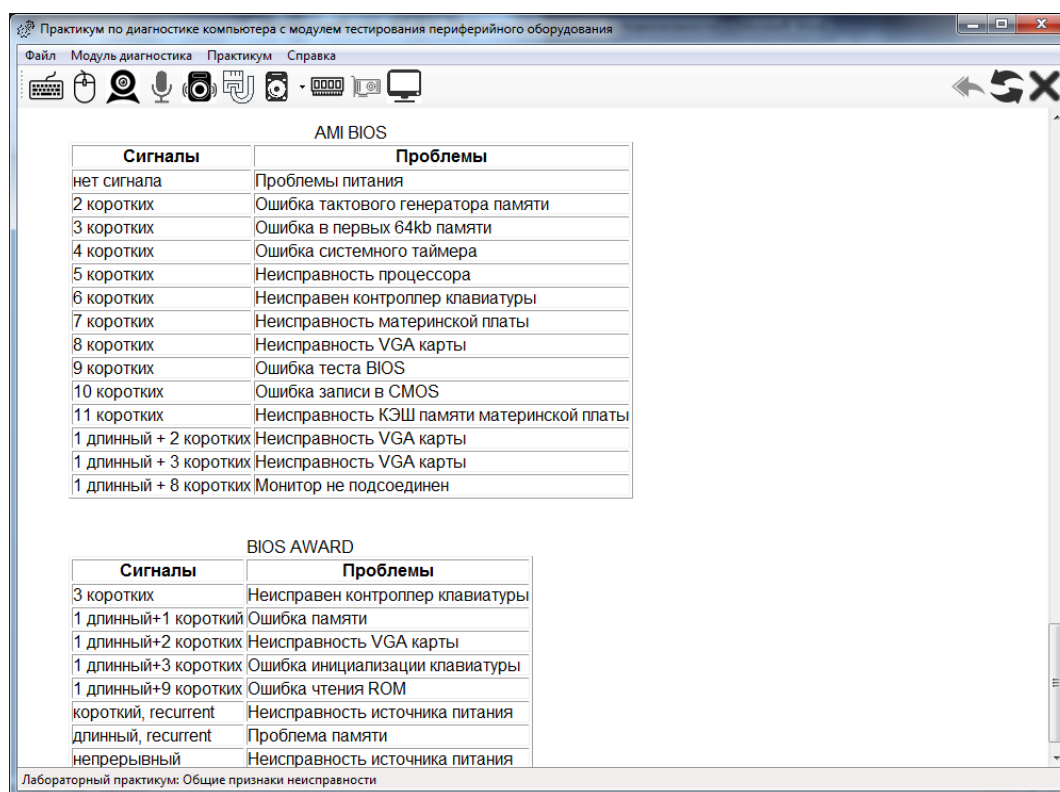


Рисунок 43 – Фрагмент №3 лабораторной работы

## Лабораторная работа №2 « Неисправности материнской платы »

**Тема:** Неисправности материнской платы.

**Цель:** Научиться производить диагностику неисправностей материнской платы, процессора персонального компьютера.

**Задачи:**

1. Научиться производить диагностику материнской платы.
2. Научиться производить тестирование процессора.
3. Научиться диагностировать блок питания.

Во второй лабораторной работе обучаемый знакомится с признаками сбоя работы материнской платы, с неисправностями процессора и средства диагностики процессора, неисправности блока питания.

Фрагменты лабораторной работы №2 продемонстрированы на рисунке 44, рисунке 45 и на рисунке 46.

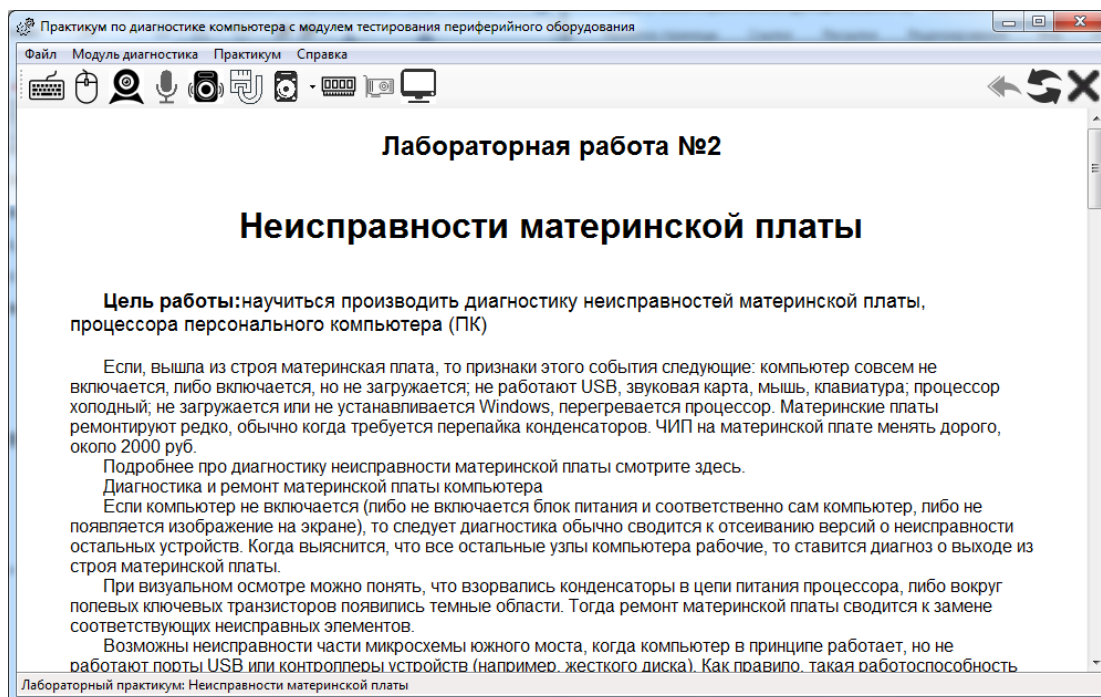


Рисунок 44 – Фрагмент №1 лабораторной работы

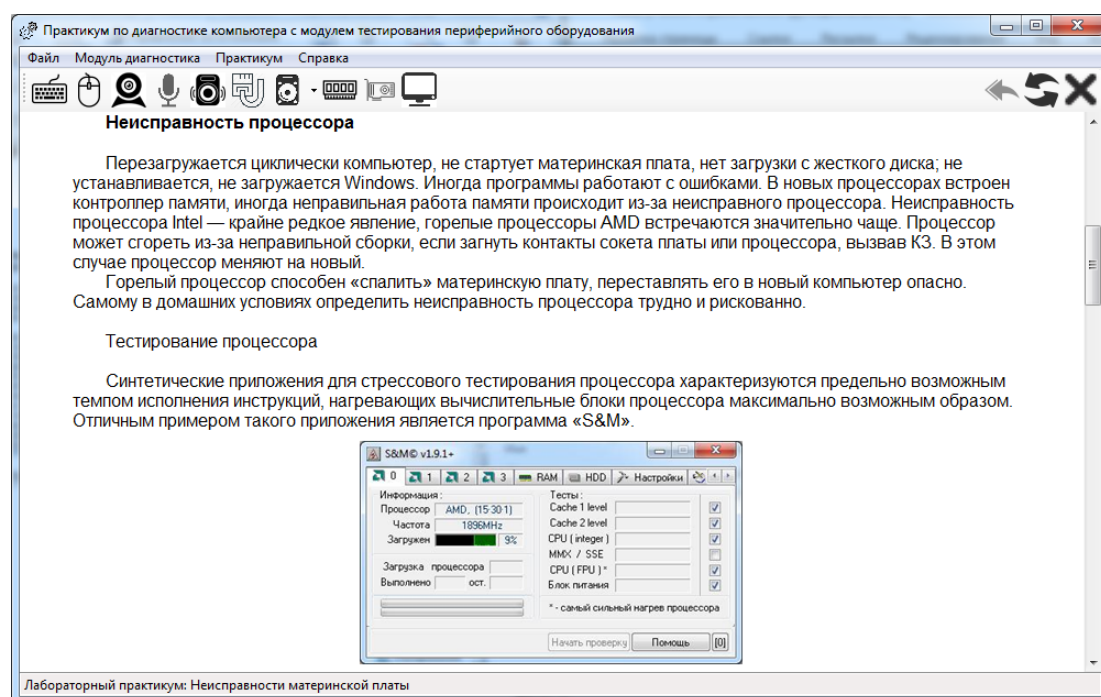


Рисунок 45 – Фрагмент №2 лабораторной работы



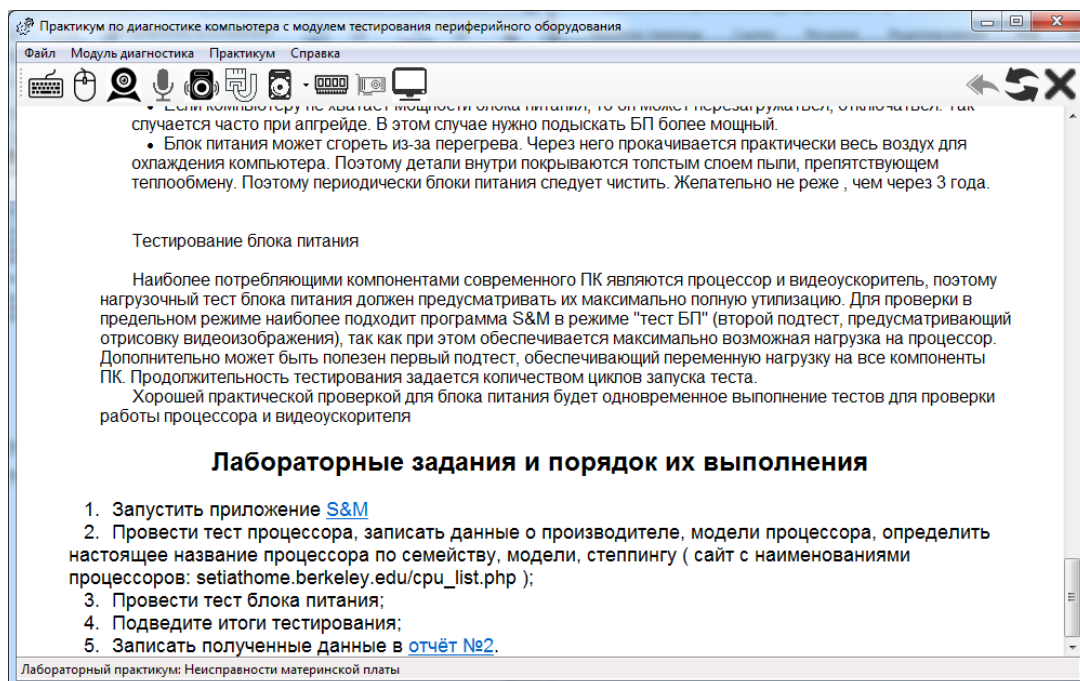


Рисунок 46 – Фрагмент №3 лабораторной работы

По завершению теоритической части лабораторной работы необходимо выполнить задания и заполнить в отчёте №2.

### Лабораторная работа №3 « Неисправности памяти »

**Тема:** Неисправности памяти.

**Цель:** Научиться производить диагностику памяти, жёсткого диска, внешних накопителей.

#### Задачи:

1. Научиться производить диагностику оперативной памяти.
2. Научиться производить тестирование оперативной памяти.
3. Научиться диагностировать неисправности жёсткого диска.
4. Научиться работать с программной диагностики CrystalDiskInfo.

В третьей лабораторной работе обучаемый знакомится с неисправностями оперативной памяти, методами и средствами его тестирования. Помимо памяти обучаемый знакомится с неисправностями жёсткого диска и программой по диагностике жёсткого диска CrystalDiskInfo.

Фрагменты лабораторной работы №3 продемонстрированы на рисунке 47, рисунке 48, рисунке 49 и на рисунке 50.

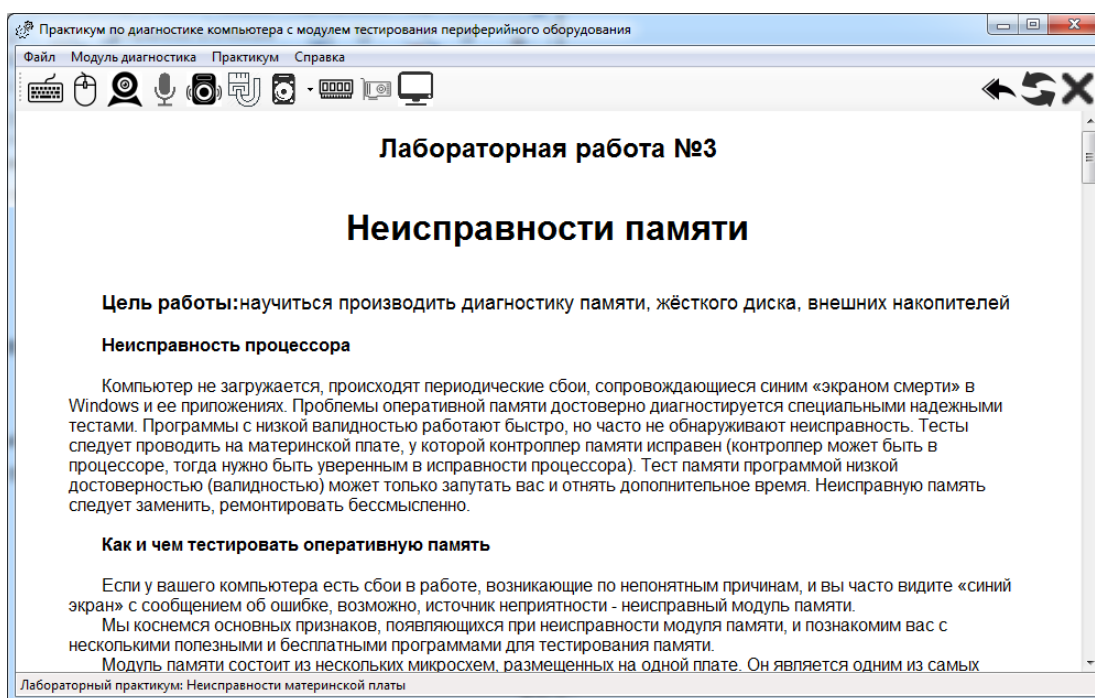


Рисунок 47 – Фрагмент №1 лабораторной работы

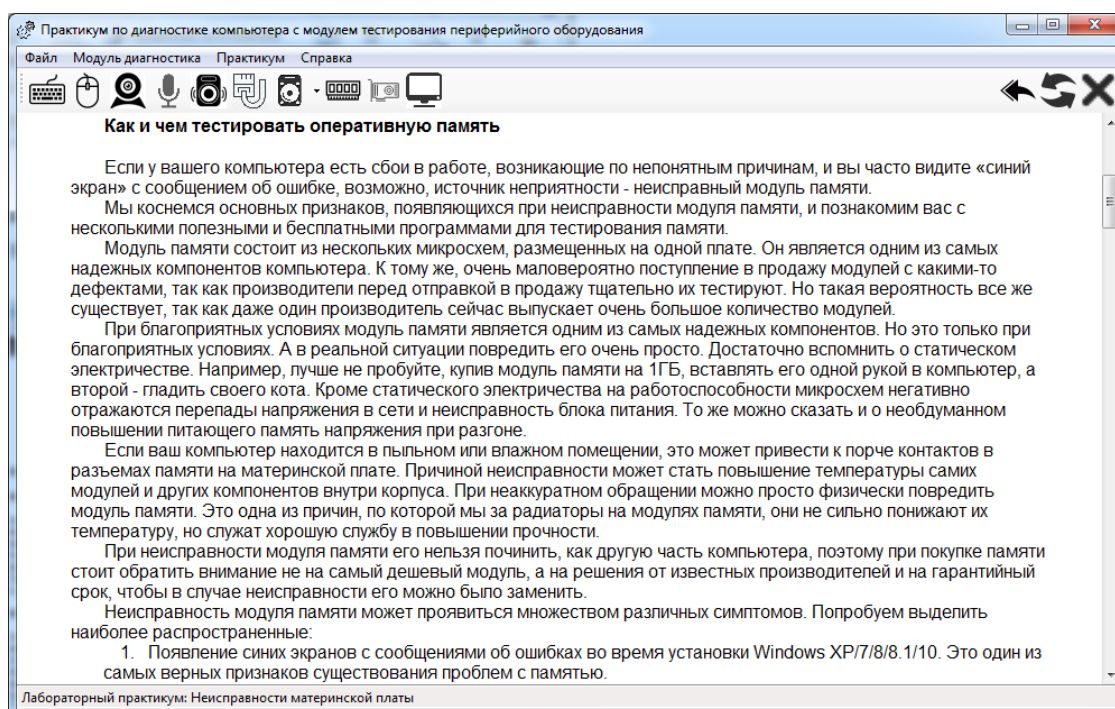


Рисунок 48 – Фрагмент №2 лабораторной работы

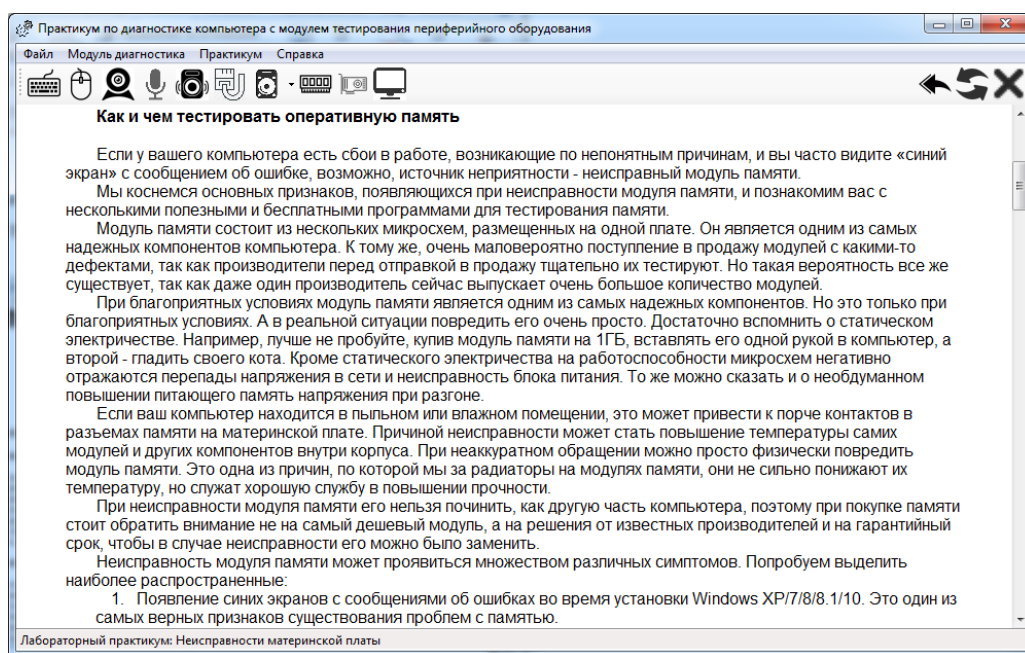


Рисунок 49 – Фрагмент №3 лабораторной работы

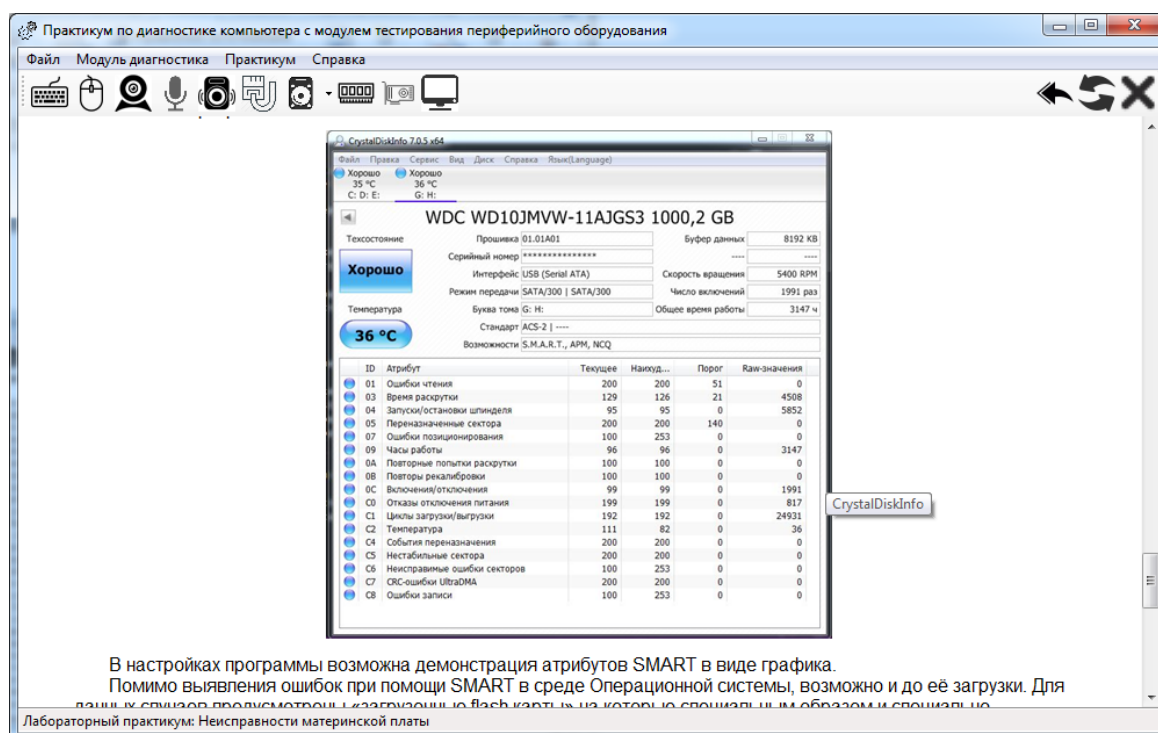


Рисунок 50 – Фрагмент №4 лабораторной работы

По завершению теоритической части лабораторной работы необходимо выполнить задания и заполнить в отчёте №3.

## Лабораторная работа №4 « Неисправности видео »

**Тема:** Неисправности видео.

**Цель:** Научиться производить диагностику неисправностей видеокарты, монитора.



## Задачи:

1. Научиться производить диагностику видеокарты компьютера.
2. Научиться производить тестирование монитора на битые пиксели.

В четвёртой лабораторной работе обучаемый знакомится с неисправностями видеокарты, методами и средствами его тестирования. Типы неисправностей монитора и способы проверки и диагностики монитора.

Фрагменты лабораторной работы №4 продемонстрированы на рисунке 51, рисунке 52 и на рисунке 53.

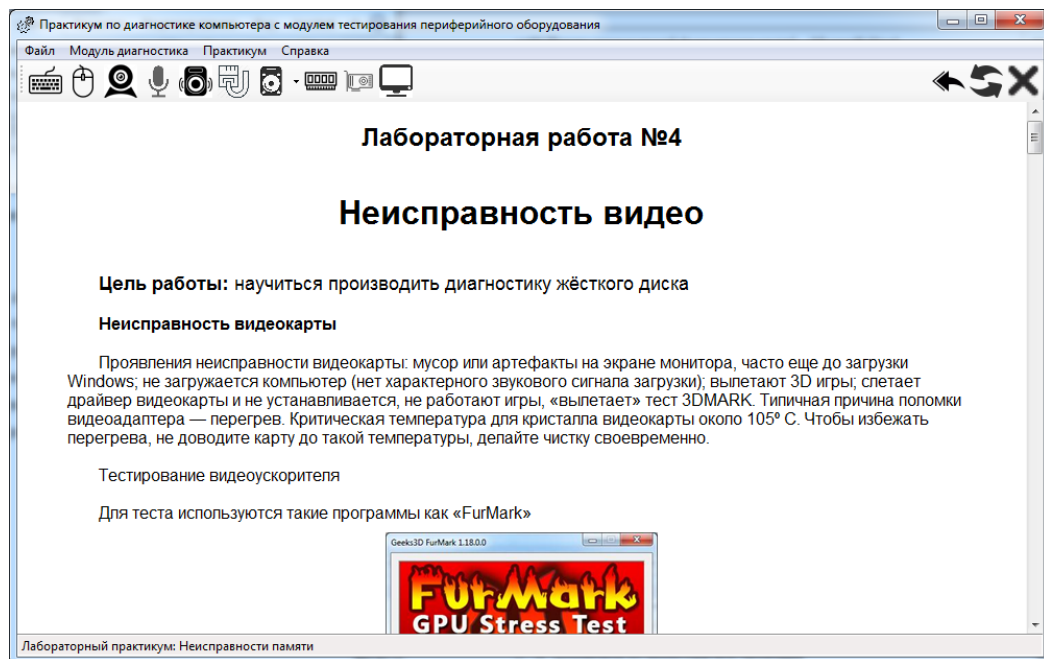


Рисунок 51 – Фрагмент №1 лабораторной работы

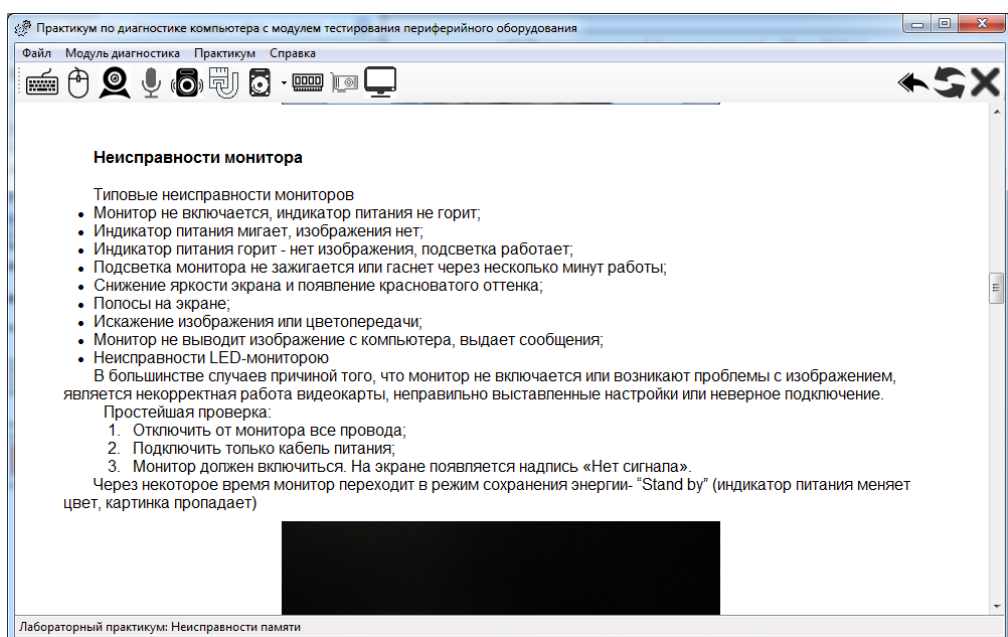


Рисунок 52 – Фрагмент №2 лабораторной работы

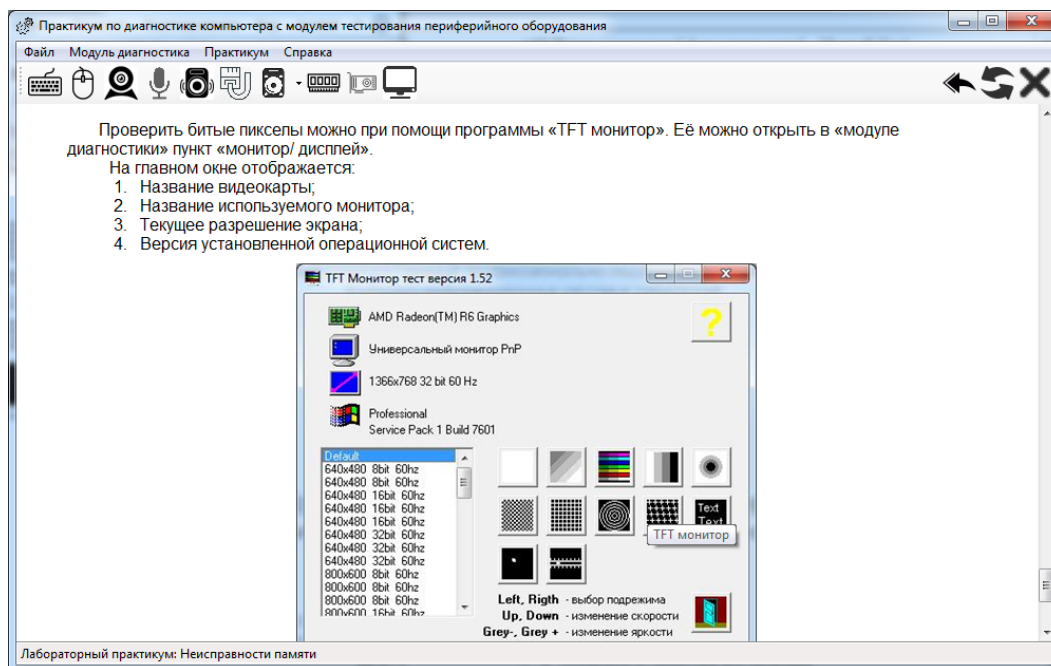


Рисунок 53 – Фрагмент №3 лабораторной работы

По завершению теоритической части лабораторной работы необходимо выполнить задания и заполнить в отчёте №4.

## **Лабораторная работа №5 « Неисправности периферийного оборудования »**

**Тема:** Неисправности видео.

**Цель:** Научиться производить диагностику периферийного оборудования персонального компьютера.

**Задачи:**

1. Научиться производить тестирование клавиатуры.
2. Научиться производить тестирование мыши.
3. Научиться производить тестирование сетевого оборудования.
4. Научиться производить тестирование аудиосистемы.
5. Научиться производить тестирование web-камеры.

В пятой лабораторной работе обучаемый знакомится со средствами диагностики и тестирования периферийного оборудования: клавиатуры, компьютерной мыши, сетевого оборудования, аудиосистемы, web-камеры.

Фрагменты лабораторной работы №5 продемонстрированы на рисунке 54, рисунке 55, рисунке 56, рисунке 57, рисунке 58.

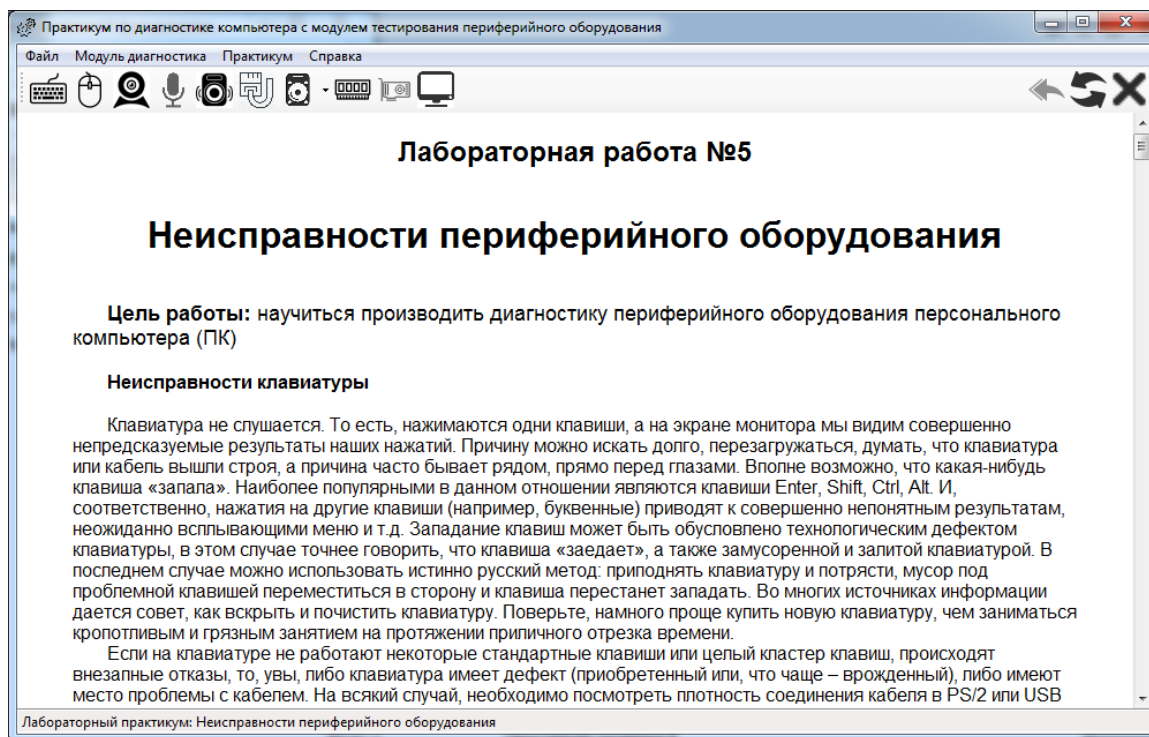


Рисунок 54 – Фрагмент №1 лабораторной работы

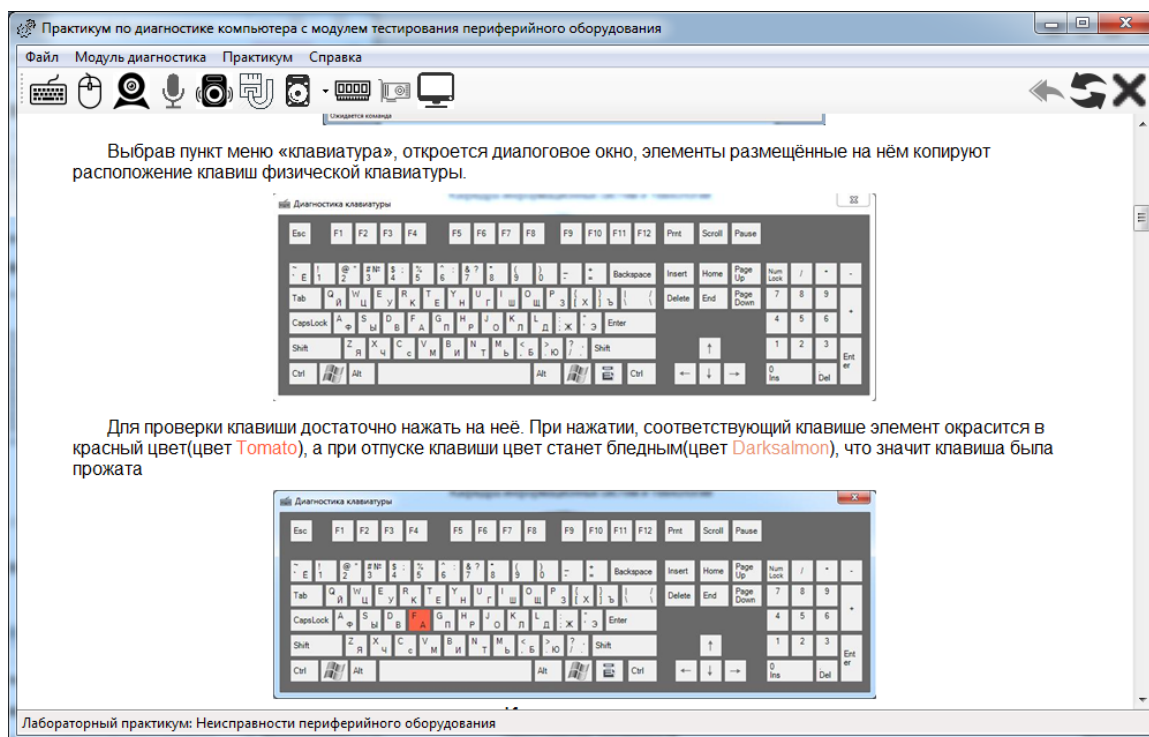


Рисунок 55 – Фрагмент №2 лабораторной работы

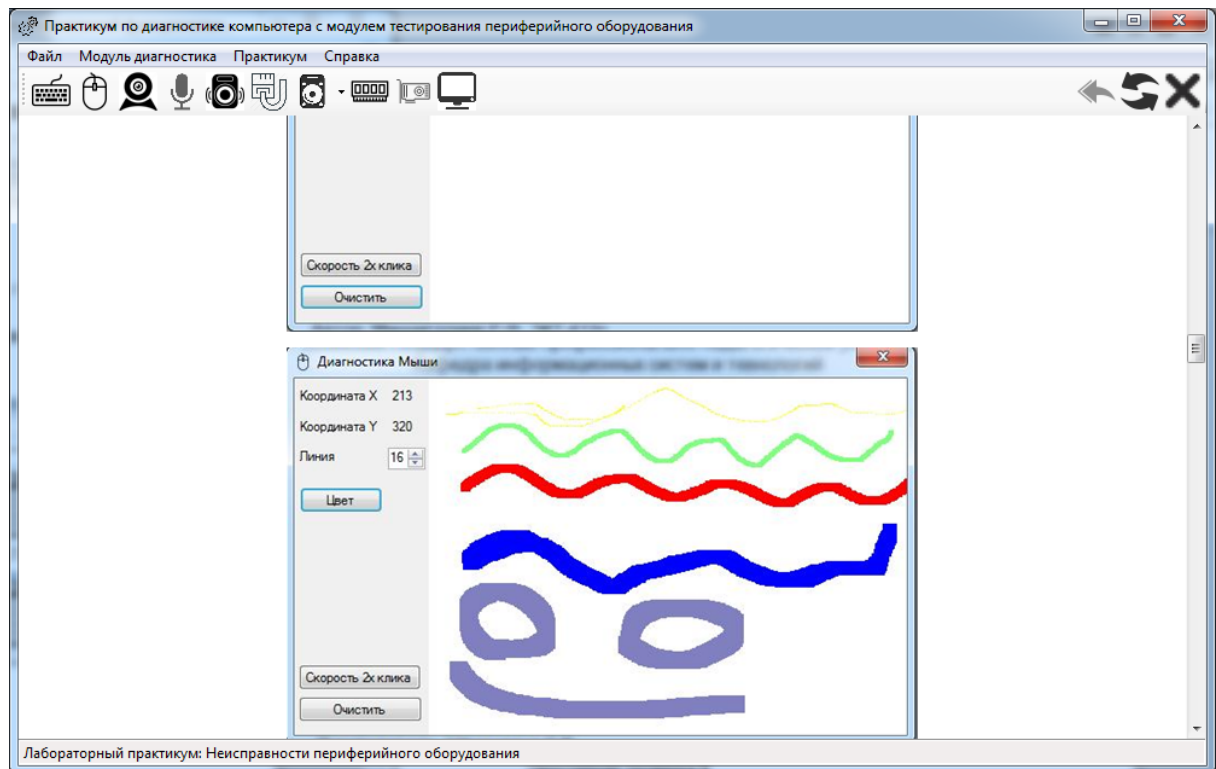


Рисунок 56 – Фрагмент №3 лабораторной работы

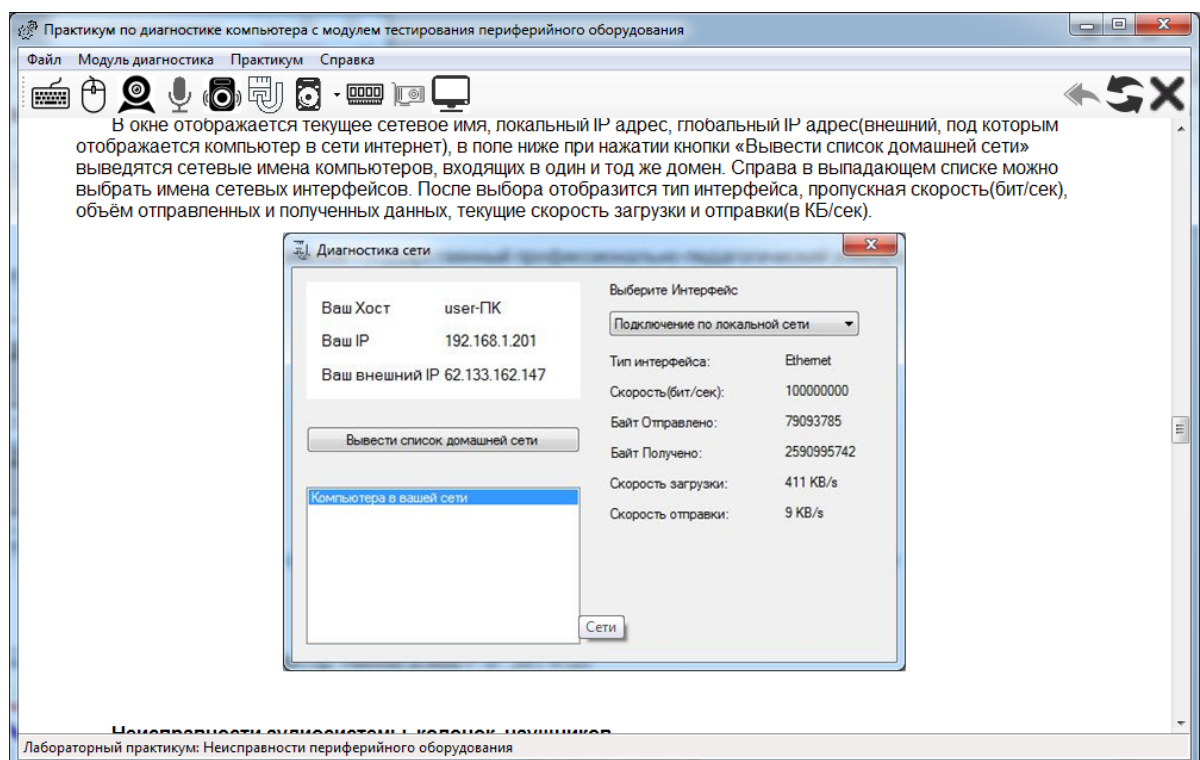


Рисунок 57 – Фрагмент №4 лабораторной работы

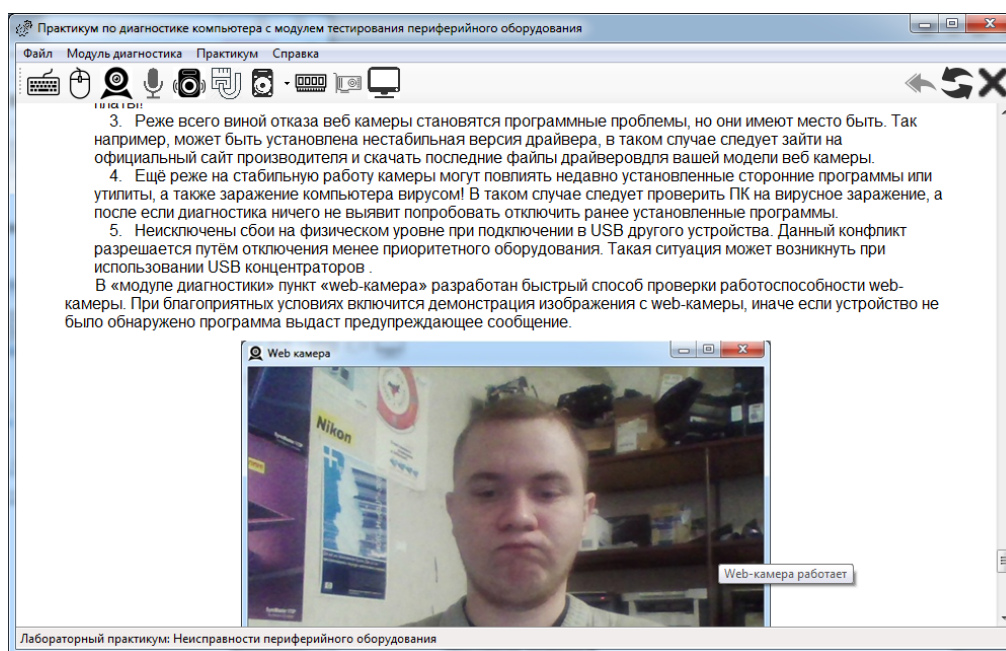


Рисунок 58 – Фрагмент №5 лабораторной работы

По завершению теоритической части лабораторной работы необходимо выполнить задания и заполнить в отчёте №5.

## 2.8 Рекомендации для обучаемых и преподавателя

Перед изучением выполнением лабораторного практикума обучающиеся должны обладать базовыми познаниями в компонентах компьютера, а так же уметь:

1. Открывать и редактировать текстовые документы.
2. Аккуратно извлекать и вставлять оборудование компьютера.
3. Работать с навигацией программы.
4. Взаимодействовать с приложениями (для некоторых необходимы права администратора).
5. Систематизировать полученные данные в тестировании и делать выводы по ним.

Рекомендации для преподавателя:

1. Подготовить рабочее место для запуска лабораторного практикума.
2. Проследить за работоспособностью компонентов требующих права администратора.

3. Сбор отчётов после выполнения лабораторных работ.
4. Проверка знаний обучающихся по всем разделам лабораторных работ при помощи программы MyTest.
5. Подведение итогов работы студентов.

## **2.9 Апробация практикума и модуля**

Лабораторный практикум были апробирован специалистами сервисного центра «общества с ограниченной ответственностью торгово-производственной компаний Техносервис». Модули диагностики запускались на 15 машинах с различными конфигурациями оборудования, на которых, были установлены операционные системы семейства Windows, начиная от версии XP, помимо Vista, и заканчивая версией 10.

По результатам апробации выяснилось, что для работы практикума необходима установка, на целевом компьютере, .Net Framework не ниже версии 4.0.

## ЗАКЛЮЧЕНИЕ

Компьютеры и компьютерная техника на данный момент времени не являются редкой или роскошной вещью. В каждом доме имеются и более одного компьютера, ноутбука или любого другого гаджета, например смартфонов, те же компьютеры только карманного вида.

Спрос рождает предложение. Компьютеры имеют свойство ломаться, выходит из строя или по забывчивости человеческой натуры не быть включенными в сеть 220 вольт, в случае с системными блоками, не имеющих встроенных аккумуляторов. Не редко даже с самой маленькой проблемой обращаются в сервисные центры, при этом потратив много средств по доставке и оплате ремонта. Ведь можно избежать растрат, подучив или почитав информацию о диагностике и ремонту: компьютеров и периферии.

Специалистам, начинающим работать в сфере оказания услуг, данный лабораторный практикум позволит укрепить знания в деле диагностики компьютеров. Тем самым улучшив навыки ремонта, и нахождения программных средств, для решения различных вопросов по диагностике оборудования.

В ходе выполнения дипломной работы был разработан электронный лабораторный практикум по теме «Практикум по диагностике компьютера и периферийного оборудования» для дисциплины «Компьютерные технологии» и для работы специалистов общества с ограниченной ответственностью торгово-производственной компаний «Техносервис».

Была проанализирована учебно-методическая документация по компьютерному строению, его диагностике и ремонту.

Материал для наполнения лабораторного практикума взят из печатных и электронных источников, что помогло наполнить его различной, но краткой, информацией.

Средствами языка разметки «HTML», средствами каскадных стилей «CSS» и средой разработок Microsoft Visual C# 2010 был разработан интер-

фейс практикума, с интуитивной навигацией и панелью быстрого доступа, и наглядностью представления материала.

Основные преимущества лабораторного практикума:

1. Информативный материал, пошагово представленный в лабораторных работах.
2. Простой интерфейс для работы.
3. Панель быстрого доступа для быстрого открытия модулей диагностики.
4. Применение практикума не ограничивается учебными заведениями, его можно применять в сервисных центрах для увеличения знаний специалистов.

В соответствии с поставленной целью в работе выполнены следующие задачи:

1. Проанализированы данные в области неисправностей и диагностики компьютерного оборудования.
2. Подготовлены лабораторные работы на основе использования программных средств компьютерной диагностики.
3. Разработан модуль диагностики периферийного оборудования.
4. Реализован интерфейс и проведено наполнение практикума с возможностью запуска средств диагностики.
5. Протестирован продукт на различных компьютерных архитектурах.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Авдеев В.А. «Организация ЭВМ и периферия с демонстрацией имитационных моделей» [Текст]: учебное пособие / В.А. Авдеев – Москва: ДМК Пресс, 2014. – 708 с.
2. Албахари Д. «С# 6.0. Справочник. Полное описание языка. 6-е изд.» [Текст] / Д. Албахари, Б. Албахари. Москва: Издательский дом Вильямс, 2016. – 1040с.
3. Бахтизин В.В. «Технология разработки программного обеспечения» [Текст]: учебное пособие для вузов / В.В. Бахтизин, Л.А. Глухова – Минск: БГУИР, 2010 – 267с.
4. Ганжа В.А. «Компьютерные сети. Введение» [Текст]: учебно-методическое пособие / В.А. Ганжа, В.В. Шиманский – Минск: БГУИР, 2015. – 155 с.
5. Захват видео с помощью OpenCV C# [Электронный ресурс]. – Режим доступа: <http://www.codeproject.com/Articles/722569/Video-Capture-using-OpenCV-with-Csharp> (дата обращения: 10.12.16).
6. Захват видео при помощи Emgu [Электронный ресурс]. – Режим доступа: [http://www.emgu.com/wiki/index.php?title=Video\\_Files](http://www.emgu.com/wiki/index.php?title=Video_Files) (дата обращения: 10.12.16).
7. Интерактивный учебник по Visual C# [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/bb383962\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/bb383962(v=vs.90).aspx) (дата обращения: 25.09.16).
8. Исходные коды. Примеры C# [Электронный ресурс]. – Режим доступа: <http://kbyte.ru/ru/Programming/Sources.aspx?mode=tech&tid=3&page=1> (дата обращения: 25.09.16).
9. Макаровский Д.М. «100 способов ускорить работу вашего компьютера» [Текст]: справочник / Д.М. Макаровский — Москва : Издательство «Э», 2016. — 224 с.

10. Неисправности клавиатуры, мыши, периферийных устройств [Электронный ресурс]. – Режим доступа: [http://mc-ps.ru/neispravnosti\\_klaviatURY\\_myshi\\_audio.html](http://mc-ps.ru/neispravnosti_klaviatURY_myshi_audio.html) (дата обращения: 16.11.16).
11. Несправности ПК и устранение их [Электронный ресурс]. – Режим доступа: <http://pk03.com.ua/article-4.html> (дата обращения: 25.01.17).
12. НОУ Интуит. Основы программирования на С# [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/2247/18/info> (дата обращения: 25.09.16).
13. НОУ Интуит. С# для школьников [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/564/420/info> (дата обращения: 25.09.16).
14. Павловская Т.А. С#. Программирование на языке высокого уровня: Учебник для вузов» [Текст]: учебник / Т.А. Павловская – Санкт-Петербург: Питер, 2009. - 432с.
15. Почему перестала работать веб камера? Проблемы с веб камерой [Электронный ресурс]. – Режим доступа: <http://wd-x.ru/problems-to-webcam/> (дата обращения: 15.01.17).
16. Практическое руководство. Воспроизведение звука в приложении [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/bb383890\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/bb383890(v=vs.90).aspx) (дата обращения: 4.01.17).
17. Пример работы с web-камерой [Электронный ресурс]. – Режим доступа: <https://blogs.msdn.microsoft.com/rucoding4fun/2010/04/27/2483/> (дата обращения: 10.12.16).
18. Проверка видеокарты на производительность, тест на стабильность [Электронный ресурс]. – Режим доступа: <http://pcpro100.info/proverka-videokartyi/> (дата обращения: 10.01.17).
19. Программирование [Электронный ресурс]. – Режим доступа: <http://www.inf1.info/book/export/html/216> (дата обращения: 25.09.16).

20. Ремонт жестких дисков компьютеров ноутбуков [Электронный ресурс]. – Режим доступа: <http://www.complace.ru/zhestkix-diskov/> (дата обращения: 20.12.16).
21. Робсон Э. «Изучаем HTML, XHTML и CSS». [Текст]: учебник / Э. Робсон, Э.Фримен. – 2-е изд. – Санкт-Петербург: Питер, 2014. — 720 с.
22. Статистика поломок ноутбуков [Электронный ресурс]. – Режим доступа: [http://allpchelp.ru/poleznoe/statistika\\_polomok\\_noutbukov/](http://allpchelp.ru/poleznoe/statistika_polomok_noutbukov/) (дата обращения: 25.01.17).
23. Уроки программирования с нуля [Электронный ресурс]. – Режим доступа: [muscsharp.ru](http://muscsharp.ru) (дата обращения: 20.11.16).
24. Устранение неисправностей и ремонт ПК своими руками на 100% [Электронный ресурс]. – Режим доступа: <http://wm-help.net/lib/b/book/2622677994/1> (дата обращения: 25.01.17).
25. Хабрахабр [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru> (дата обращения: 25.09.16).
26. Хейлсберг А. «Язык программирования C#. Классика Computers Science. 4-е изд.» [Текст]: учебное пособие / А. Хейлсберг, М.Торгерсен, С. Вилтамут и др. под руководством А. Хейлсберг. – Санкт-Петербург: Питер, 2012. — 784 с.
27. CyberForum.ru - форум программистов и сисадминов [Электронный ресурс]. – Режим доступа: <http://www.cyberforum.ru> (дата обращения: 25.09.16).
28. C#. Берём камеру под контроль [Электронный ресурс]. – Режим доступа: <http://www.vr-online.ru/content/c-berem-web-kameru-pod-kontrol-2417>(дата обращения: 10.12.16).
29. Memtest: RAM reliability tester for Windows [Электронный ресурс]. – Режим доступа: <http://hcidesign.com/memtest/> (дата обращения: 25.12.16).

# ПРИЛОЖЕНИЕ 1

**Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования**

**«Российский государственный профессионально-педагогический университет»**  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий  
направление 44.03.04 Профессиональное обучение (по отраслям)  
профиль «Информатика и вычислительная техника»  
профилизация «Компьютерные технологии»

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ Н. С. Толстова

« 16 » \_\_\_\_\_ ноября \_\_\_\_\_ 2016 г.

**ЗАДАНИЕ  
на выполнение выпускной квалификационной работы бакалавра**

студента 4 курса, группы ЗКТ-412С Миннигалиева Рустама Фавиловича

1. Тема «Практикум по диагностике компьютера с модулем тестирования периферийного оборудования» утверждена распоряжением по институту от 16.11.2016 г. № 68/1.
2. Руководитель Ченушкина Светлана Владимировна, ст. преподаватель кафедры ИСиТ
3. Место преддипломной практики ООО ТПК «Техносервис»
4. Исходные данные к ВКР  
Хабрахабр [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru>  
CyberForum.ru - форум программистов и сисадминов [Электронный ресурс]. – Режим доступа: <http://www.cyberforum.ru>  
Интерактивный учебник по Visual C# [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/ru-ru/library/bb383962\(v=vs.90\).aspx](https://msdn.microsoft.com/ru-ru/library/bb383962(v=vs.90).aspx)  
Неисправности ПК и устранение их [Электронный ресурс]. – Режим доступа: <http://pk03.com.ua/article-4.html>
5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)  
Анализ в области неисправностей и диагностики компьютерного оборудования  
Подготовка лабораторных работ на основе использования программных средств компьютерной диагностики  
Разработка модуля диагностики периферийного оборудования  
Реализация интерфейса практикума
6. Перечень демонстрационных материалов  
Презентация выполненная в Microsoft PowerPoint

## 7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа дипломной работы	Срок выполнения этапа	Процент выполнения ВКР	Отметка руководителя о выполнении
1	Сбор информации по выпускной работе и сдача зачета по преддипломной практике	09.01.2017	15	
2	Выполнение работ по разрабатываемым вопросам их изложение в выпускной работе:			
	Анализ литературы, интернет-источников	12.12.2016	10	
	Подготовка лабораторных работ по теме ВКР	03.12.2016	15	
	Разработка модуля диагностики	07.12.2016	15	
	Реализация интерфейса и наполнение практикума	09.01.2017	15	
	Тестирование продукта	12.01.2017	10	
3	Оформление текстовой части ВКР	15.01.2017	5	
4	Выполнение демонстрационных материалов к ВКР	30.01.2017	5	
5	Нормоконтроль	02.02.2017	5	
6	Подготовка доклада к защите в ГЭК	08.02.2017	5	

## 8. Консультанты по разделам выпускной квалификационной работы

Наименование раздела	Консультант	Задание выдал		Задание принял	
		подпись	дата	подпись	дата

Руководитель \_\_\_\_\_  
подпись                      дата

Задание получил \_\_\_\_\_  
подпись студента                      дата

9. Выпускная квалификационная работа и все материалы проанализированы. Считаю возможным допустить Миннигалиева Р.Ф. к защите выпускной квалификационной работы в государственной экзаменационной комиссии.  
Руководитель \_\_\_\_\_  
подпись                      дата

10. Допустить Миннигалиева Р.Ф. к защите выпускной квалификационной работы в государственной экзаменационной комиссии (протокол заседания кафедры от 08.02.17 №7)  
Заведующий кафедрой \_\_\_\_\_  
подпись                      дата

## ПРИЛОЖЕНИЕ 2

### Синтаксис «MainFrm.cs»:

```
namespace Diag
{
    public partial class MainFrm : Form
    {
        string tsslbltext = "Ожидается команда";
        string ProccesName;
        public string FileHtml;
        string PractTitleName = "";
        int MainFrmX;
        int MainFrmY;
        public MainFrm()
        {
            InitializeComponent();
        }
        private void TSSlblDefault()
        {
            TSSlbl1.Text = tsslbltext;
            this.Activate();
        }

        private void menuItem6_Click(object sender, EventArgs e)
        {
            TSSlbl1.Text = "Запущена диагностика клавиатуры";
            KeyBoardFrm keybrdfm = new KeyBoardFrm();
            keybrdfm.ShowDialog();
            TSSlblDefault();
        }
        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            menuItem6_Click(sender, e);
        }
        private void menuItem7_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start(@"webcamtest\DiagWeb.exe");
            ProccesName = "DiagWeb";
            TmrProccess.Enabled = true;
        }
        private void menuItem9_Click(object sender, EventArgs e)
        {
            TSSlbl1.Text = "Диагностика Сети";
            TCPFrm tcpfrm = new TCPFrm();
```

```

tcpfrm.ShowDialog();
TSSlblDefault();
}
private void menuItem8_Click(object sender, EventArgs e)
{
    TSSlbl1.Text = "Диагностика Микрофона";
    MicroFrm microfrm = new MicroFrm();
    microfrm.ShowDialog();
    TSSlblDefault();
}
private void menuItem12_Click(object sender, EventArgs e)
{
    TSSlbl1.Text = "Диагностика Speaker";
    SpeakerFrm speakerfrm = new SpeakerFrm();
    speakerfrm.ShowDialog();
    TSSlblDefault();
}
private void menuItem13_Click(object sender, EventArgs e)
{
    TSSlbl1.Text = "Диагностика Мыши";
    MouseFrm mousefrm = new MouseFrm();
    mousefrm.ShowDialog();
    TSSlblDefault();
}
public bool procesOn = false;
string procname = "";
private void TmrProccess_Tick(object sender, EventArgs e)
{
    if (procesOn == false)
    {
        procname = ProccesName;
        switch (procname)
        {
            case "DiagWeb":
                TSSlbl1.Text = "Запущена диагностика Web-камеры";
                break;
            case "DiskInfo32":
                TSSlbl1.Text = "Запущена диагностика Жёсткого диска";
                break;
            case "DiskInfo64":
                TSSlbl1.Text = "Запущена диагностика Жёсткого диска";
                break;
            case "memtest":
                TSSlbl1.Text = "Запущена диагностика Оперативной памяти";
                break;
            case "FurMark":
                TSSlbl1.Text = "Запущена диагностика Видеокарты/видеочипа";

```

```

        break;
    case "tftmonitor":
        TSSlbl1.Text = "Запущена диагностика Монитора/ Дисплея";
        break;
    default:
        MessageBox.Show("Процесс не обнаружен");
        break;
    }
}
if (procname != "")
{
    bool b = Process.GetProcessesByName(procname).Any();
    if (b == true)
    {
        this.Enabled = false;
        proccesOn = true;
    }
    else
    {
        proccesOn = false;
        ProccesName = "";
        TSSlblDefault();
        this.Enabled = true;
        TmrProcces.Enabled = false;
    }
}
}
private void menuItem20_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start(@"tftmonitor\tftmonitor.exe");
    ProccesName = "tftmonitor";
    TmrProcces.Enabled = true;
}
private void menuItem2_Click(object sender, EventArgs e)
{
    this.Close();
}
private void toolStripButton1_Click_1(object sender, EventArgs e)
{
    menuItem13_Click(sender, e);
}
private void menuItem14_Click(object sender, EventArgs e)
{
    FileHtml = "motheboard.html";
    PractTitleName = menuItem14.Text;
    WebPraktikum();
}

```



```

    }
    private void menuItem6_Click_1(object sender, EventArgs e)
    {
        FileHtml = "pc.html";
        PractTitleName = menuItem6.Text;
        WebPraktikum();
    }
    string str = "";
    string lasturl;
    private void WebPraktikum()
    {
        string path = Application.StartupPath + @"\" + zadaniya + FileHtml;
        lasturl = path;
        if (File.Exists(path))
        {
            this.Size = new System.Drawing.Size(1000, 528);

            toolStripButton5.Visible = true;
            toolStripButton4.Visible = true;
            toolStripButton6.Visible = true;
            webBrowser1.Navigate(path);
            Stream aStream = webBrowser1.DocumentStream;
            if (aStream != null)
            {
                Encoding hDocEncoding = Encoding.GetEncoding(webBrowser1.Document.Encoding);
                StreamReader SR = new StreamReader(aStream, hDocEncoding);
                do
                {
                    str = str + SR.ReadLine() + "\n";
                } while (SR.Peek() >= 0);
            }
            webBrowser1.Visible = true;
            TmrLect.Enabled = true;
        }
        else
        {
            MessageBox.Show("Лабораторный практикум не был обнаружен");
        }
    }
    private void MainForm_Load(object sender, EventArgs e)
    {
        this.Size = new System.Drawing.Size(1000, 628);
        MainFormX = this.Width;
        MainFormY = this.Height;
        this.MinimumSize = new System.Drawing.Size(1000, 628);
        TSSlbl1.Text = tsslbltext;
    }

```

```

private void MainFrm_Resize(object sender, EventArgs e)
{
    int Cntr = Convert.ToInt32(Math.Round( Convert.ToDecimal(this.Width/2)));
    int CntrY = Convert.ToInt32(Math.Round(Convert.ToDecimal(this.Height/2)));
    label1.Location = new System.Drawing.Point(lblpos(label1.Width, Cntr), label1.Location.Y);
    label2.Location = new System.Drawing.Point(lblpos(label2.Width, Cntr), label2.Location.Y);
    pictureBox1.Location = new System.Drawing.Point(lblpos(pictureBox1.Width, Cntr), pictureBox1.Location.Y);
    label3.Location = new System.Drawing.Point( (Cntr - label3.Width) , label3.Location.Y);
}

private int lblpos(int lblwidth, int cntrfrm)
{
    int NewPosX;
    int Cntrlbl = Convert.ToInt32(Math.Round(Convert.ToDecimal(lblwidth/ 2)));
    NewPosX = cntrfrm - Cntrlbl;
    return NewPosX;
}

private void toolStripButton4_Click_1(object sender, EventArgs e)
{
    webBrowser1.Visible = false;
    toolStripButton4.Visible = false;
    toolStripButton5.Visible = false;
    toolStripButton6.Visible = false;
    TmrLect.Enabled = false;
    MainFrm_Load(sender, e);
    lasturl = "";
    FileHtml = "";
}

private void MainFrm_MinimumSizeChanged(object sender, EventArgs e)
{
    MainFrm_Load(sender, e);
}

private void menuItem7_Click_1(object sender, EventArgs e)
{
    FileHtml = "ram.html";
    PractTitleName = menuItem7.Text;
    WebPraktikum();
}

private void menuItem8_Click_1(object sender, EventArgs e)
{
    FileHtml = "video.html";
    PractTitleName = menuItem8.Text;
    WebPraktikum();
}

private void menuItem9_Click_1(object sender, EventArgs e)
{
    FileHtml = "periphery.html";
    PractTitleName = menuItem9.Text;
}

```

```

        WebPraktikum();
    }
    private void toolStripButton5_Click_1(object sender, EventArgs e)
    {
        webBrowser1.Refresh();
    }
    private void TmrLect_Tick(object sender, EventArgs e)
    {
        string LectFile = lasturl;
        if (webBrowser1.Url.LocalPath.ToString() != LectFile)
        {
            toolStripButton6.Enabled = true;
            TmrLect.Enabled = false;
        }
        else
        {
            TSSlbl1.Text = "Лабораторный практикум: "+ webBrowser1.DocumentTitle.ToString();
            toolStripButton6.Enabled = false;
        }
    }

    private void toolStripButton6_Click(object sender, EventArgs e)
    {
        WebPraktikum();
    }
    private void menuItem5_Click(object sender, EventArgs e)
    {
        about about = new about();
        about.ShowDialog();
    }
}
}

```

Синтаксис « KeyBoardFrm.cs»:

```

namespace Diag
{
    public partial class KeyBoardFrm : Form
    {
        public KeyBoardFrm()
        {
            InitializeComponent();
        }
        Color Tomato = Color.Tomato;
        Color Salmon = Color.DarkSalmon;
        [DllImport("user32.dll")]
        private static extern short GetAsyncKeyState(Keys vKey);
    }
}

```

```

private void mainFrm_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.ShiftKey)
    {
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.LShiftKey)))
        {
            lblShiftLeft.BackColor = Salmon;
        }
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.RShiftKey)))
        {
            lblShiftRight.BackColor = Salmon;
        }
    }
    if (e.KeyCode == Keys.ControlKey)
    {
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.LControlKey)))
        {
            lblCtrlLeft.BackColor = Salmon;
        }
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.RControlKey)))
        {
            lblCtrlRight.BackColor = Salmon;
        }
    }
    if (e.KeyCode == Keys.Menu)
    {
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.LMenu)))
        {
            lblAltLeft.BackColor = Salmon;
        }
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.RMenu)))
        {
            lblAltRight.BackColor = Salmon;
        }
    }

    if (e.KeyCode == Keys.LWin)
    {
        lblWinLeft.BackColor = Tomato;
    }
    else if (e.KeyCode == Keys.RWin)
    {
        lblWinRight.BackColor = Tomato;
    }
    if (e.KeyCode == Keys.Left)
    {
        lbl37.BackColor = Tomato;
    }
}

```

```

    }
    else if (e.KeyCode == Keys.Up)
    {
        lbl38.BackColor = Tomato;
    }
    else if (e.KeyCode == Keys.Right)
    {
        lbl39.BackColor = Tomato;
    }
    else if (e.KeyCode == Keys.Down)
    {
        lbl40.BackColor = Tomato;
    }
    int but = Convert.ToInt32(e.KeyCode);
    e.SuppressKeyPress = true;
    string lblSname = "lbl" + but.ToString();
    foreach (Control ctr in this.Controls)
    {
        if (ctr is Label)
        {
            if (ctr.Name == lblSname)
            {
                ctr.BackColor = Tomato;
                break;
            }
        }
    }
}

private void mainFrm_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.ShiftKey)
    {
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.LShiftKey)))
        {
            lblShiftLeft.BackColor = Salmon;
        }
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.RShiftKey)))
        {
            lblShiftRight.BackColor = Salmon;
        }
    }
    if (e.KeyCode == Keys.ControlKey)
    {
        if (Convert.ToBoolean(GetAsyncKeyState(Keys.LControlKey)))
        {
            lblCtrlLeft.BackColor = Salmon;
        }
    }
}

```

```

    }
    if (Convert.ToBoolean(GetAsyncKeyState(Keys.RControlKey)))
    {
        lblCtrlRight.BackColor = Salmon;
    }
}
if (e.KeyCode == Keys.Menu)
{
    if (Convert.ToBoolean(GetAsyncKeyState(Keys.LMenu)))
    {
        lblAltLeft.BackColor = Salmon;
    }
    if (Convert.ToBoolean(GetAsyncKeyState(Keys.RMenu)))
    {
        lblAltRight.BackColor = Salmon;
    }
}
if (e.KeyCode == Keys.LWin)
{
    lblWinLeft.BackColor = Salmon;
}
else if (e.KeyCode == Keys.RWin)
{
    lblWinRight.BackColor = Salmon;
}
if (e.KeyCode == Keys.Left)
{
    lbl37.BackColor = Salmon;
}
else if (e.KeyCode == Keys.Up)
{
    lbl38.BackColor = Salmon;
}
else if (e.KeyCode == Keys.Right)
{
    lbl39.BackColor = Salmon;
}
else if (e.KeyCode == Keys.Down)
{
    lbl40.BackColor = Salmon;
}

}
int but = Convert.ToInt32(e.KeyCode);
string lblSname = "lbl" + but.ToString();
foreach (Control ctr in this.Controls)
{
    if (ctr is Label)

```

```

        {
            if (ctr.Name == lblSname )
            {
                ctr.BackColor = Salmon;
                break;
            }
        }
    }
}
}
}
}
}

```

#### Синтаксис «MicroFrm.cs»:

```

namespace Diag
{
    public partial class MicroFrm : Form
    {
        WaveInEvent waveIn = new WaveInEvent();
        public MicroFrm()
        {
            InitializeComponent();
            btnMic.Text = "Начать тест";
        }
        private void WaveOnDataAvailable(object sender, WaveInEventArgs e)
        {
            for (int index = 0; index < e.BytesRecorded; index += 2)
            {
                short sample = (short)((e.Buffer[index + 1] << 8) | e.Buffer[index + 0]);
                float amplitude = sample / 32768f;
                float level = Math.Abs(amplitude); // от 0 до 1
                ((VolumeMeter)this.Controls["VolumeMeter1"]).Amplitude = level;
            }
        }
        private void btnMic_Click(object sender, EventArgs e)
        {
            if (btnMic.Text == "Начать тест")
            {
                btnMic.Text = "Завершить тест";
                waveIn.DataAvailable += WaveOnDataAvailable;
                waveIn.WaveFormat = new WaveFormat(8000, 1);
                waveIn.StartRecording();
            }
            else if (btnMic.Text != "Начать тест")
            {
                btnMic.Text = "Начать тест";
                waveIn.StopRecording();
                waveIn.Dispose();
            }
        }
    }
}

```

```

        waveIn = null;
        ((VolumeMeter)this.Controls["VolumeMeter1"]).Amplitude = 0;
    }
}
}
}

```

### Синтаксис «MouseFrm.cs»:

```

namespace Diag
{
    public partial class MouseFrm : Form
    {
        bool bdraw = false;
        Color clr = new Color();
        public MouseFrm()
        {
            InitializeComponent();
        }
        private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
        {
            if (clr.Name == "0")
                clr = Color.Black;
            lblXinfo.Text = e.X.ToString();
            lblYinfo.Text = e.Y.ToString();
            if (bdraw == true)
            {
                Graphics graphics = pictureBox1.CreateGraphics();
                graphics.FillRectangle(new SolidBrush(clr), e.X, e.Y, Convert.ToInt32(numericUpDown1.Value), Convert.ToInt32(numericUpDown1.Value));
            }
        }
        private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
        {
            if (e.Button.ToString() == "Left")
            {
                bdraw = true;
            }
            else if (e.Button.ToString() == "Right")
            {
                bdraw = false;
            }
        }
        private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
        {
            bdraw = false;
        }
        private void btnClear_Click(object sender, EventArgs e)
    }
}

```



```

{
    Graphics graphics = pictureBox1.CreateGraphics();
    graphics.Clear(Color.White);
}
private void BtnColor_Click(object sender, EventArgs e)
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
        clr = colorDialog1.Color;
}
private void btnSpeed_Click(object sender, EventArgs e)
{
    PnlClickDiag.Enabled = true;
    PnlClickDiag.Visible = true;
    label1.Visible = true;
    label2.Visible = true;
}
private void PnlClickDiag_MouseDoubleClick(object sender, MouseEventArgs e)
{
    if (startCl == true)
    {
        startCl = false;
        string date1 = DateTime.Now.ToString("fff");
        int a = Convert.ToInt32(date);
        int b = Convert.ToInt32(date1);
        if(b<a)
        {
            b += 1000;
        }
        label1.Text = (Math.Abs(b-a)).ToString();
        // kol = 0;
    }
}
string date;
bool startCl = false;
private void PnlClickDiag_Click(object sender, EventArgs e)
{
    if (startCl == false)
    {
        startCl = true;
        date = DateTime.Now.ToString("fff");
    }
}
}
}

```

Синтаксис «SpeakerFrm.cs»:

namespace Diag

```

{
    public partial class SpeakerFrm : Form
    {
        SoundPlayer player = new SoundPlayer();
        public SpeakerFrm()
        {
            InitializeComponent();
            PicBxLeft.Image = Properties.Resources.kol1;
            PicBxRight.Image = Properties.Resources.kol1;
            btnAudio.Text = "Начать тест";
        }
        private void btnAudio_Click(object sender, EventArgs e)
        {
            playleft();
            Thread.Sleep(2000);
            TimerKol.Enabled = true;
        }
        private void leftkol()
        {
            PicBxLeft.Image = Properties.Resources.kol2;
        }
        private void playleft()
        {
            leftkol();
            player.Stream = Properties.Resources.muztestl;
            player.LoadAsync();
            player.Play();
            player.Dispose();
            PicBxLeft.Image = Properties.Resources.kol1;
        }
        private void playright()
        {
            player.Stream = Properties.Resources.muztestr;
            player.LoadAsync();
            player.Play();
            player.Dispose();
            Thread.Sleep(2000);
            PicBxRight.Image = Properties.Resources.kol1;
        }
        private void TimerKol_Tick(object sender, EventArgs e)
        {
            PicBxLeft.Image = Properties.Resources.kol1;
            PicBxRight.Image = Properties.Resources.kol2;
            playright();
            TimerKol.Enabled = false;
        }
    }
}

```